2008

Artificial neural network analysis of the mechanical properties of tungsten fiber/bulk metallic glass matrix composites via neutron diffraction and finite element modelling

Baris Denizer
Iowa State University

Artificial neural network analysis of the mechanical properties of tungsten fiber/bulk metallic glass matrix composites via neutron diffraction and finite element modeling

by

Baris Denizer

A thesis submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Major: Material Science and Engineering

Program of Study Committee:
Ersan Ustundag, Co-major Professor
Halil Ceylan, Co-major Professor
Michael R. Kessler

Iowa State University

Ames, Iowa

2008

To my wife Eimile

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGEMENTS

Thanks first to Dr. Ersan Ustundag, my major professor, for his guidance and support throughout this research and the writing of this thesis. His high expectations motivated me to accomplish more than I had imagined.

I am grateful to Dr. Halil Ceylan, my co-major professor, for the discussions we had on artificial neural networks. His expertise enabled me to perform interdisciplinary research on a subject which was totally new to me. I would also like to thank Dr. Michael Kessler for joining my committee.

I would like to give special thanks to Dr. Seung-Yub Lee for helping me with finite element analysis and providing me with deep insight into bulk metallic glasses. At the times when I was frustrated he helped me to focus on my research.

Here I would like to extend my thanks to my research group colleagues Mesut Varlioglu, Goknur Tutuncu, Sarah Shiley Haubrich, Li Li and Youngshin Kim for helping me with various aspects of conducting research. I would also like to thank Yusuf Yusufoglu for his friendship and reading early versions of my thesis.

I am eternally grateful to my parents Dundar and Filiz and, my brother Cagri for their encouragement and life-long support.

Last but not least, I would like to express my thanks to my wife, Eimile. Her love and support not only helped me to focus on my work but also made my life a lot more enjoyable during my last year at Iowa State.

This work was partially supported by the DANSE software project developed under NSF award DMR-0520547.

ABSTRACT

The deduction of mechanical properties from experimental data is essentially an inverse analysis where the data are compared to the predictions of a model by modifying the input parameters for the model until a satisfactory match is attained. Often times, this is done manually via trial-and-error. There are, however, rigorous mathematical methods that offer robust inverse analyses with enhanced accuracy. In the present study, artificial neural networks (ANN) are employed to deduce the in-situ constitutive laws of tungsten (W) fiber reinforced bulk metallic glass (BMG) matrix composites.

Experimental data consist of lattice strain from the W fibers obtained by neutron diffraction and total composite strain measured by an extensometer. The mechanical behavior of the composites is modeled via finite element analysis (FEA). The constitutive behavior of the fibers and the matrix are described using the Voce and power laws, respectively. The effect of thermal residual stresses is also included as a function of freezing temperature below which, residual stresses start to build up during cooldown.

The goal of the present inverse analysis via ANN is to optimize the values of the Voce and power law parameters plus the freezing temperature a total of seven parameters. First, a forward ANN is constructed that attempts to match the predictions of FEA which is run multiple times via a random selection of the seven parameters. Next, inverse ANN are constructed to optimize the values of the seven parameters, i.e., as inverse models. Finally, the optimized parameters are input to the forward ANN and compared to the experimental data.

This is the first application of ANN in the analysis and interpretation of engineering diffraction data. It demonstrates the power of ANN in conducting robust inverse analysis of such data. The approach developed and presented here can also be employed in the optimization

of engineering diffraction experiments to increase their accuracy and efficiency.

CHAPTER 1. INTRODUCTION

1.1 Mechanical Modeling of W/BMG Composites

Mechanical behavior of composite materials depends on the in-situ mechanical response of the constituent phases. Materials within a composite body often behave differently than their monolithic forms and a complete understanding of the in-situ mechanical behavior of each phase is necessary in order to model the mechanical behavior of the composite material.

The present study employed W-fiber-reinforced bulk metallic glass, BMG-matrix composites as a model system. These have been studied extensively before [2]; so, extensive data exists about their mechanical behavior. These composites also effectively illustrate the need to perform in-situ experiments (such as neutron diffraction) to measure the response of each phase to external loading. The additional problem here is the fact that the amorphous BMG matrix does not allow "lattice strain" measurement via diffraction (with adequate accuracy). Therefore, a mechanical model is needed to fully interpret the data and deduce the constitutive behavior of all phases.

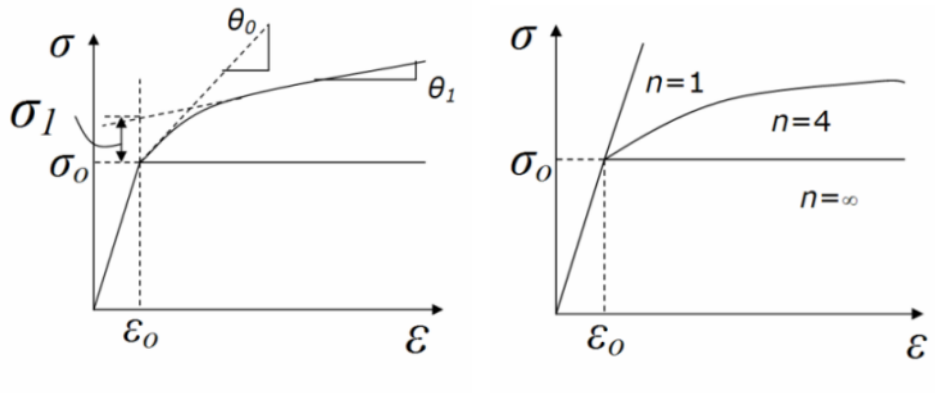
In modeling the mechanical behavior of W fiber reinforced BMG matrix composites Voce [3] and power laws are found to give the best fit for the in-situ mechanical response of the tungsten fibers and the BMG matrix respectively. [4] Voce law is given by

$$\sigma = \sigma_0 + (\sigma_1 + \theta_1 \epsilon)\left(1 - \exp\left(-\frac{\theta_0 \epsilon}{\sigma_1}\right)\right) \tag{1.1}$$

Power law is given by

$$\begin{aligned} \frac{\epsilon}{\epsilon_0} &= \frac{\sigma}{\sigma_0} && \sigma \leq \sigma_0 \\ \frac{\epsilon}{\epsilon_0} &= \left(\frac{\sigma}{\sigma_0}\right)^n && \sigma > \sigma_0 \end{aligned} \tag{1.2}$$

Figure 1.1 describes Voce and power law parameters.



(a) Voce law (b) Power law

Figure 1.1 Graphical representation of Voce and power laws

W/BMG composites are prepared by melt infiltration casting in which, cleaned and straightened tungsten fibers inserted in a quartz-glass tube at 1200 $^{\circ}C$ are infiltrated with molten BMG ($Zr_{41.2}Ti_{13.8}Cu_{12.5}Ni_{10.0}Be_{22.5}$) and then quenched in a brine solution. [2] The thermal residual stresses build up during this process affect the mechanical behavior of the composite significantly and therefore, they must be taken into account in modeling. The residual stresses are calculated as a function of the "freezing" temperature below which, the stresses build up during cooldown. [5]

1.2 Current Solution

Estimation of the constitutive behavior of W/BMG composites requires knowledge of the "freezing" temperature and, Voce and power law parameters. Various optimization techniques can be used to estimate these parameters to fit loading-unloading curves obtained from experiments or simulations. For instance, previous work on the same composites employed trial-and-error for this purpose. [6] However, this is not a rigorous process and cannot handle multiple material parameters with complex (and unknown) interrelationships. Once trained and tested,

neural networks make accurate predictions in shorter times than conventional optimization techniques.

1.3 Proposed Solution

Voce and power laws are used to model the constitutive behavior of tungsten and BMG respectively and the freezing temperature is used to calculate the residual stresses induced in the composite material during cooling. Therefore, optimum values of the Voce and power law parameters and the "freezing" temperature – a total of seven parameters – are required to define the constitutive behavior of the fibers, matrix and composite. A framework of artificial neural networks is developed to optimize these parameters and to plot the corresponding stress-strain curves. The framework consists of forward and inverse neural networks. Both networks are trained with data from finite element modeling by using the back-propagation algorithm. The forward neural network model maps the functional relationship between the seven parameters and the strains that evolve in the material upon a loading-unloading cycle. After successfully trained the forward neural network model can be used to plot the stress-stain curves accurately in shorter times than finite element modeling. The inverse neural network model gives the optimized values of the seven parameters governing the constitutive behavior of the phases and the composite. The massively parallel interconnected architecture of the neural network enables fast and accurate optimization. These two neural network models are combined to form a framework which not only performs inverse analysis but also offers an alternative approach to finite element modeling.

The finite element model (FEM) serves two important functions here: (i) It allows the estimate of load sharing between the fibers and the matrix and also yields the composite behavior. (ii) It permits the simulation of thousands of datasets that would be impossible to generate experimentally, yet are crucial for the successful training of neural networks. Therefore, the reader should always keep in mind a crucial assumption in this study: namely that the FEM presented here together with the mathematical equations employed in the Voce and power laws are accurate representations of the real behavior of the phases and the composite

Table 1.1 Ranges of Voce and power law parameters, and the freezing temperature

Variable	**Unit**	**Min**	**Max**
σ_0^W	MPa	1000	1500
σ_1	MPa	500	800
θ_0	MPa	600000	900000
θ_1	MPa	1200	1450
σ_0^{BMG}	MPa	1900	2100
n	–	2	15
Temperature	°C	200	390

First, finite element modeling is used to generate a database of loading-unloading curves for the fiber elastic strain and the composite total strain by randomly changing the functional variables and the "freezing" temperature in the ranges given in Table 5.2. These ranges are determined in an earlier study by Ustundag et al. [4] The database is used to train and test the forward and the inverse neural network models. Python programming language is used to control the commercial finite element modeling software $ABAQUS^{TM}$ to generate the database. The three dimensional finite element model involves a hexagonal quarter symmetry unit cell loaded under uniaxial compression with plane strain assumed in the axial direction. Second order 20 node brick elements with reduced integration points are used to model the cylindrical fibers. Figure 1.2 shows the quarter symmetry finite element model for the 20 % composite and Table 1.2 shows the properties of the tungsten fibers and the BMG matrix [6].

Table 1.2 Properties of the BMG matrix and the tungsten fibers

Material	**Young's Modulus (GPa)**	**Poisson's Ratio**	**CTE** ($^\circ C^{-1}$)
BMG	96	0.36	$9.0x10^-6 - 15.0x10^-6$
Tungsten	410	0.28	$4.5x10^-6 - 4.7x10^-6$

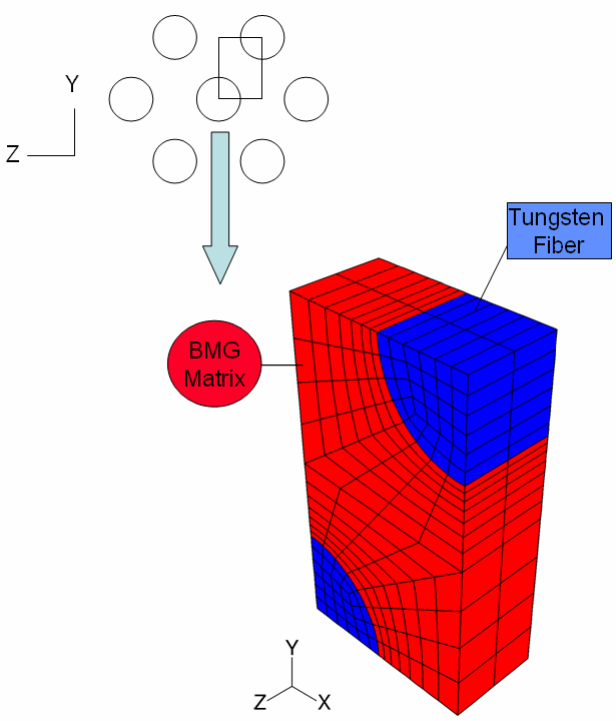


Figure 1.2 Hexagonal quarter symmetry finite element model

A sensitivity study is conducted to evaluate the influence of each parameter on the constitutive behavior of the fibers and the composite by finite element analysis. A Python script is used to control $ABAQUS^{TM}$ to set six of the variables to their corresponding mean values while changing the value of the analyzed variable between its minimum and maximum at five evenly spaced points. The loading-unloading curves obtained are plotted in series for the fibers and the composite separately to observe the change in the constitutive behavior of the materials. Results of the sensitivity study are instrumental in developing the forward and the inverse neural network models.

A forward neural network model is developed to predict the loading-unloading curves of the fibers and the composite and then the neural network predictions are validated by the loading-unloading curves obtained from the finite element model. The inputs to the neural network are Voce parameters (σ_0^W, σ_1, θ_0 and θ_1), power law parameters (σ_0^{BMG} and power law exponent) and freezing temperature. The outputs from the network are the strains in the

fibers and in the composite at the data points on their respective loading-unloading curves. After successfully trained and tested, the forward neural network model maps the functional relationship between the input parameters and the output parameters. This enables prediction of the fiber and the composite stress-strain curves without the need for finite element analysis and in shorter times than otherwise would be obtained by finite element modeling.

Inverse neural network models are developed to predict the optimum values of Voce and power law parameters and the freezing temperature. In this case a separate neural network model is constructed to predict each of the variables. The inputs to the neural networks are the strains at the data points on the loading-unloading curves of the fibers and the composite and each of the neural network models output the optimized value of one of the variables.

The predicted optimized functional parameters and the freezing temperature are fed to the forward neural network model and the finite element model developed earlier and the results obtained are compared to each other and the experimental loading-unloading curves. The match between the loading-unloading curves obtained from the finite element model and the forward neural network model proves the reliability of the developed neural network framework. On the other hand the match between the experimental loading-unloading curve and the curve obtained from the finite element model validates the finite element model which is used to train the neural networks. In other words, if the curve obtained from finite element model matches the experimental data well, then the curve obtained from the neural network framework is also expected to match to the experimental data well.

Finally, the results obtained from the inverse analysis are further refined by iterative optimization using the neural network framework where necessary.

The framework developed enables planning of experimental procedures for in-situ and macroscopic mechanical testing of composite materials as well as offering an alternative modeling approach to finite element modeling. The neural network framework developed is the first application of coupled forward-inverse neural network models in the field of Engineering Diffraction.

1.4 Thesis Overview

The first chapter briefly presented the proposed research. The second chapter explains the theory of artificial neural networks with a special emphasis on multilayer feed forward neural networks and the back-propagation algorithm. The third chapter reviews the published literature in the use of neural networks in materials science and mechanical model development. The fourth chapter presents a sensitivity study conducted to analyze the influence of each parameter on the constitutive behavior of the fibers and the composite. The fifth chapter presents the forward model and the results obtained in this study. The sixth chapter discusses results from the inverse model. The seventh chapter presents the iterative optimization approach adopted to refine the inverse neural network predictions. Chapter eight presents the conclusions.

CHAPTER 2. ARTIFICIAL NEURAL NETWORKS

2.1 Fundamentals

2.1.1 Introduction

Artificial neural networks use a parallel distributed computation paradigm alternative to the sequential instruction based computation. It is inspired by the structure of real neural networks in the human brain and the knowledge acquired from neuroscience is used to develop robust, noise and fault-tolerant computation architectures capable of generalization. [7]

A contemporary definition of artificial neural networks is adapted from [8] as follows:

Any computing architecture consisting of massively parallel interconnected simple neural processors is called an artificial neural network.

The above definition suggests simple neural processors, which are often called neurons in literature, are central to the implementation of artificial neural networks. A mathematical model of a neuron is presented next.

2.1.2 Mathematical Model of a Neuron

Fig. 2.1 shows a model of a neuron in the form of a *directed graph*. The neuron is stimulated with n inputs through its *synaptic connections*. Each synaptic connection has a *weight*, which is a positive or a negative number showing the *strength* of that connection. Each input is multiplied with the weight of the synaptic connection and the weighted inputs are summed together with an externally applied *bias* to give the *induced local field* v_j. On a directed graph the bias can be represented as a synaptic weight whose input is always set to $+1$. Then an *activation* or *squashing* function $\varphi_j(.)$ is applied to the induced local field of the neuron, which

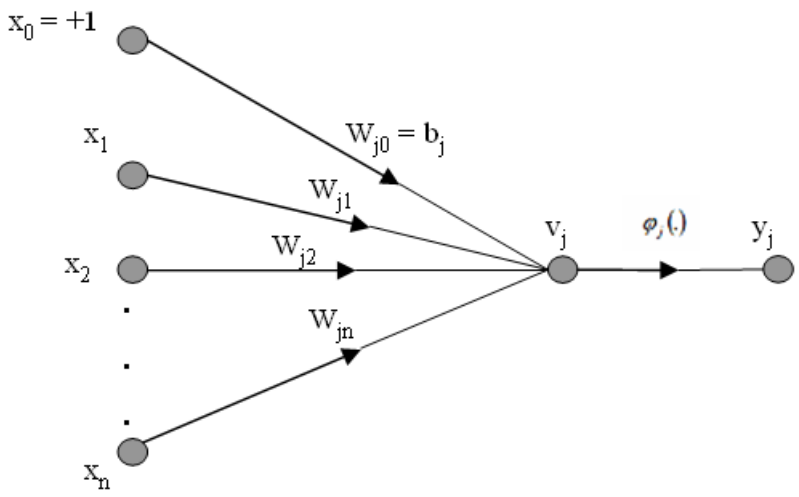


Figure 2.1 Signal flow graph of a neuron. [1]

gives the output y_j of neuron j. The activation function squashes the range of the output to a certain interval. These concepts can be explained in mathematical terms as follows:

The input vector x_i is defined as

$$x_i = [x_1, x_2 \cdots x_n] \tag{2.1}$$

and the weight vector w_{ji} is defined as

$$w_{ji} = [w_{j1}, w_{j2} \cdots w_{jn}] \tag{2.2}$$

The induced local field v_j is obtained by

$$v_j = \sum_{i=0}^{n} w_{ji} x_i \tag{2.3}$$

where n is the number of inputs and the bias b_j is given by

$$b_j = w_{j0} x_0 \tag{2.4}$$

Finally, the output y_j is obtained by

$$y_j = \varphi_j(v_j) \tag{2.5}$$

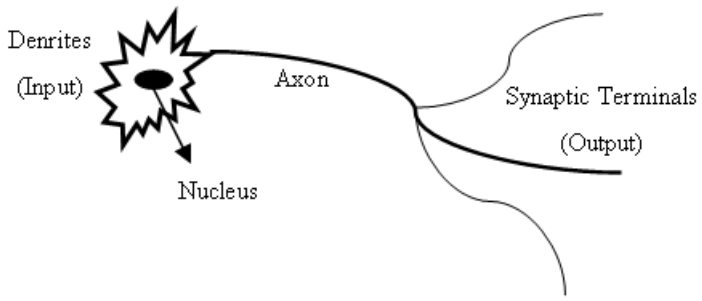


Figure 2.2 Schematic representation of a biological neuron

2.1.3 Biological Analogy

Artificial neural networks are inspired by biological neural networks. While designing artificial neural networks the structure of the human brain is simulated in a very simplified manner. Consequently, there are similarities between the mathematical model of an artificial neuron and the structure of a biological neuron. [9]

Fig. 2.2 shows a simplified structure of a biological neuron. If we compare it with Fig. 2.1 we notice the resemblance between:

- the dendrites of a biological neuron and the input nodes of an artificial neuron
- the synaptic terminals of a biological neuron and the output nodes of the artificial neuron
- the weighted nature of the synaptic connections of a biological and an artificial neuron
- the nucleus of a biological neuron; and the bias, induced local field and activation function of an artificial neuron
- the massively parallel interconnected architecture of biological and artificial neural networks

However, there are fundamental differences between the working principles of biological and artificial neurons, which are beyond the scope of this text. Of course, the structure of the human brain is not yet completely understood.

2.1.4 Network Architectures

In neural network terminology architecture refers to *the arrangement of neurons in layers and pattern of synaptic connections between them within and between layers.* [10] In a layered network there is an input layer of source nodes, an output layer of computational nodes, *neurons*, and optionally one or more hidden layers of neurons between the input and output layers. There are two main classes of neural network architectures depending on the type of connections between neurons.

2.1.4.1 Feed-Forward Neural Networks

In feed-forward neural networks the signal flows in one direction between the layers and there are no connections between the neurons in the same layer.

2.1.4.2 Recurrent Neural Networks

In recurrent neural networks there is at least one feedback loop, which can contain a unit delay operator resulting in a dynamical system and both kinds of connections are allowed.

2.1.5 Learning Paradigms

From a neural network point of view *learning is a processes in which the synaptic weights of the network are modified depending on the input.* [1] The type of learning determines how these modifications are made. There are two principal learning paradigms in neural networks.

2.1.5.1 Supervised Learning

In supervised learning a training example consisting of an input and a desired output is presented to the network. Then the synaptic weights of the network are modified in such a way as to minimize the difference between the actual network output and the desired output. [11]

2.1.5.2 Unsupervised Learning

In this case no desired output is presented. The synaptic weights of the network are corrected until the network output provides a useful representation of the input statistics. [12]

2.2 Multilayer Perceptrons

2.2.1 Architecture of a Multilayer Perceptron

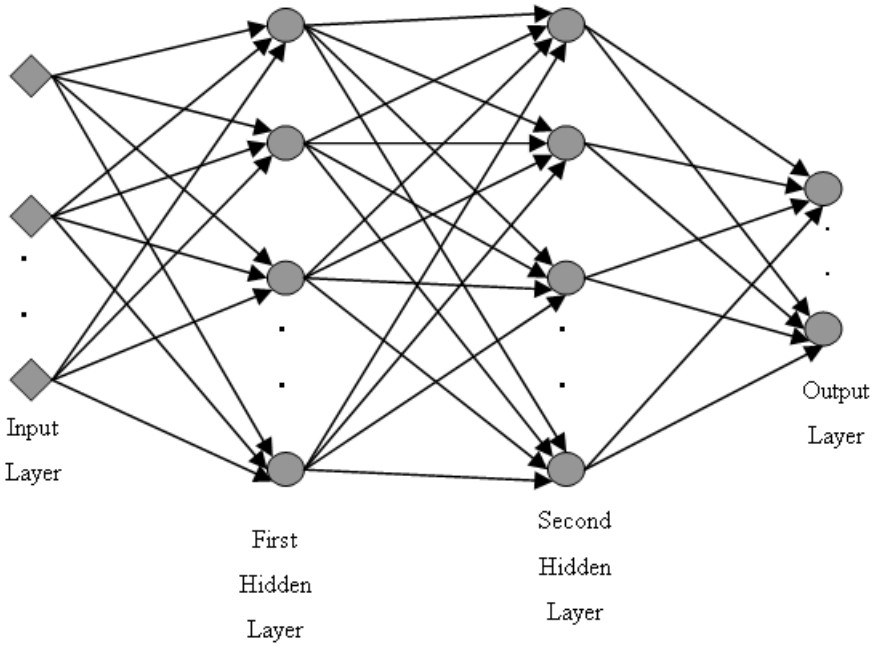


Figure 2.3 Architectural graph of a multilayer perceptron with two hidden layers. [1]

A multilayer feed-forward network, or multilayer perceptron (MLP) as commonly referred in literature, consists of an input layer, one or more hidden layers and an output layer. Fig. 2.3 illustrates a multilayer perceptron with k nodes in the input layer, l_1 neurons in the first hidden layer, l_2 neurons in the second hidden layer and m neurons in the output layer. This architecture is denoted as a k-l_1-l_2-m network. Note that the network shown here is fully connected, which means each neuron in the network is connected to a node/neuron in the

previous layer. However, there are no connections between the nodes/neurons within the same layer of the network.

Training of a multilayer perceptron is accomplished in a supervised manner by using the back-propagation algorithm. The concept of error-correction learning and the back-propagation algorithm are discussed next.

2.2.2 Error-Correction Learning

In the beginning of the training process synaptic weights of the network are assigned random values. Then the network is presented with a *training example* at each iteration. A training example consists of an input and a corresponding target. The training examples are obtained from experimental or simulation data. The network generates an output by processing the input and compares the output with the target. The difference between the target and the output determines the error. Then the synaptic weights of the network are modified by the training algorithm proportional to the error. The goal of the training process is to reduce the error below a predetermined value on an iterative basis. This requires a presentation of many training examples, which constitutes a *training set*. The presentation of a complete training set is called an *epoch*. This form of supervised learning is called *error correction learning*. Figure 2.4 shows a schematic representation of the error-correction learning.

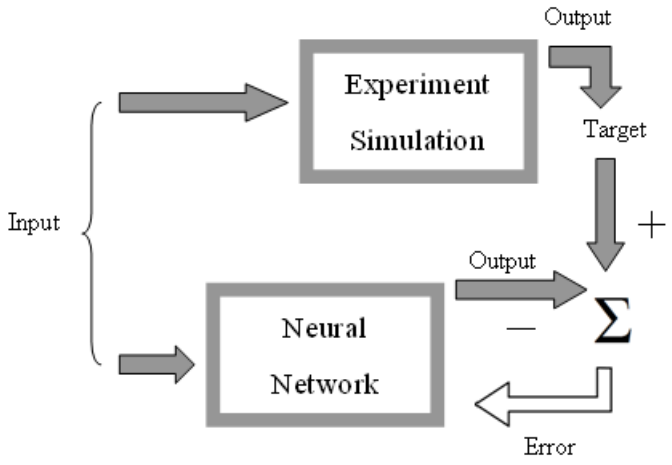


Figure 2.4 Schematic representation of error-correction learning

2.2.3 Back-Propagation Algorithm

Back-propagation algorithm is based on the error correction learning rule. In the words of Rumelhart and his coworkers [13]

> *The application of the generalized delta rule involves two phases: During the first phase the input is presented and propagated forward through the network to compute the output value for each unit* [neuron]. *This output is than compared to the targets, resulting in an error signal for each output unit. The second phase involves a backward pass through the network during which the error signal is passed to each unit in the network and the appropriate weight changes are made.*

The weight changes are made according to the method of steepest gradient descent. The local gradients are calculated depending on the position of a neuron within the network. Derivation of the back-propagation algorithm for a neuron in an output layer and a hidden layer is described in detail below. [1]

2.2.3.1 Case I

When neuron j is located in the output layer of the network it is provided with a target value $d_j(n)$. Consequently, its error signal $e_j(n)$ and the local gradient $\delta_j(n)$ can be calculated directly as follows: the induced local field $v_j(n)$ appearing at the input of the activation function associated with neuron j is given by

$$v_j(n) = \sum_{i=0}^{m} w_{ji}(n) y_i(n) \tag{2.6}$$

where m is the total number of inputs (excluding the bias) applied to neuron j. The synaptic weight w_{j0}, which corresponds to the fixed input $y_0 = +1$, equals the bias b_j applied to neuron j. The output signal $y_j(n)$ appearing at the output of neuron j at iteration n is given by

$$y_j(n) = \varphi_j(v_j(n)) \tag{2.7}$$

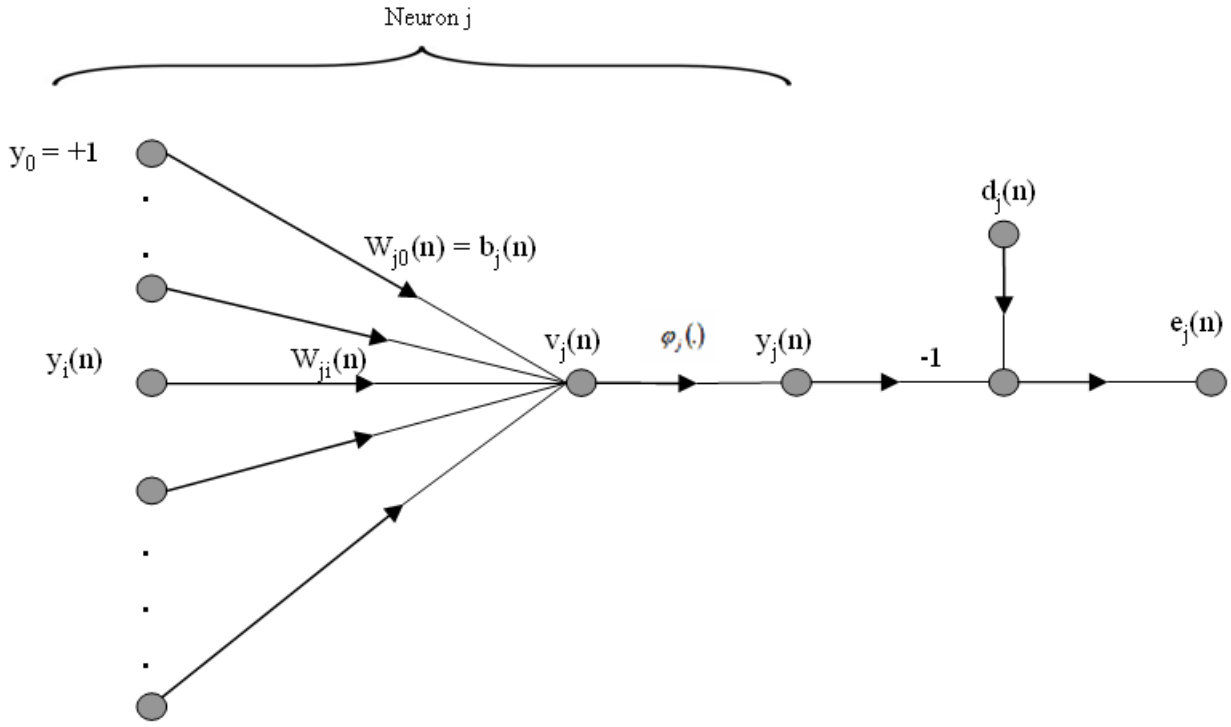


Figure 2.5 Signal-flow graph highlighting the details of output neuron j [1]

where $\varphi_j(.)$ denotes the nonlinear *activation function* applied to neuron j. The error signal at the output of neuron j at iteration n is calculated as

$$e_j(n) = d_j(n) - y_j(n) \tag{2.8}$$

The total error energy for neuron j at iteration n is given by

$$\xi(n) = \frac{1}{2}\sum_{j \in C} e_j^2(n) \tag{2.9}$$

where set C includes all the neurons in the output layer of the network. Let N denote the total number of examples in the training set. Then the average squared error energy is obtained by

$$\xi_{av} = \frac{1}{N}\sum_{n=1}^{N} \xi(n) \tag{2.10}$$

The goal of the training process is to minimize ξ_{av} by adjusting the synaptic weights of the neural network depending on the errors computed for each training example. Δw_{ji}, the synaptic correction applied to the synaptic weight $w_{ji}(n)$, which is proportional to the partial

derivative $\partial\xi(n)/\partial w_{ji}(n)$ is calculated by using the "chain rule" of calculus as follows:

$$\frac{\partial\xi(n)}{\partial w_{ji}(n)} = \frac{\partial\xi(n)}{\partial e_j(n)}\frac{\partial e_j(n)}{\partial y_j(n)}\frac{\partial y_j(n)}{\partial v_j(n)}\frac{\partial v_j(n)}{\partial w_{ji}(n)} \tag{2.11}$$

Differentiating Eq. 2.9 with respect to $e_j(n)$ we get

$$\frac{\partial\xi(n)}{\partial e_j(n)} = e_j(n) \tag{2.12}$$

Differentiating Eq. 2.8 with respect to $y_j(n)$ we get

$$\frac{\partial e_j(n)}{\partial y_j(n)} = -1 \tag{2.13}$$

Next, differentiating Eq. 2.7 with respect to $v_j(n)$ we get

$$\frac{\partial y_j(n)}{\partial v_j(n)} = \varphi_j'(v_j(n)) \tag{2.14}$$

Finally, differentiating Eq. 2.6 with respect to $w_{ji}(n)$ we get

$$\frac{\partial v_j(n)}{\partial w_{ji}(n)} = y_i(n) \tag{2.15}$$

The use of Eqs. 2.12 to 2.15 in Eq. 2.11 yields

$$\frac{\partial\xi(n)}{\partial wji(n)} = -e_j(n)\varphi_j'(v_j(n))y_i(n) \tag{2.16}$$

The correction $\Delta w_{ji}(n)$ applied to $w_{ji}(n)$ is defined by the *delta rule*:

$$\Delta w_{ji}(n) = -\eta\frac{\partial\xi(n)}{\partial w_{ji}(n)} \tag{2.17}$$

where η is the *learning rate parameter*. Learning rate parameter provides a step size for the correction made to the synaptic weight of a connection. [14] The minus sign in Eq. 2.17 accounts for *gradient descent* in weight space. The gradient of the error energy function, $\partial\xi(n)/\partial w_{ji}(n)$, shows the direction in which the function increases most rapidly. On the contrary, the negative of the gradient shows the direction in which the function decreases most rapidly. [10] Accordingly, the use of Eq. 2.16 in 2.17 yields

$$\Delta w_{ji}(n) = \eta\delta_j(n)y_i(n) \tag{2.18}$$

where the *local gradient* $\delta_j(n)$ is defined by

$$\begin{aligned}\delta_j(n) &= -\frac{\partial \xi(n)}{\partial v_j(n)} \\ &= \frac{\partial \xi(n)}{\partial e_j(n)} \frac{\partial e_j(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial v_j(n)} \\ &= e_j(n)\varphi_j'(v_j(n))\end{aligned} \tag{2.19}$$

2.2.3.2 Case II

When neuron j is located in a hidden layer of the network it is not directly provided with a target value. Therefore, the error signal for a hidden neuron can only be calculated recursively in terms of the error signals of all the output neurons to which that hidden neuron is connected. This makes computation of the local gradient for a hidden neuron complicated. According to Eq. 2.19, we may redefine the local gradient $\delta_j(n)$ for hidden neuron j as

$$\delta_j(n) = -\frac{\partial \xi(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial v_j(n)} \tag{2.20}$$

Note that the partial derivative $\partial y_j(n)/\partial v_j(n)$ can be calculated by using Eq. 2.14. Hence, Eq. 2.20 becomes

$$\delta_j(n) = -\frac{\partial \xi(n)}{\partial y_j(n)} \varphi_j'(v_j(n)) \tag{2.21}$$

To calculate the partial derivative $\partial \xi(n)/\partial y_j(n)$, we may proceed as follows. From Eq. 2.7 we see that

$$\xi(n) = \frac{1}{2} \sum_{k \in C} e_k^2(n) \tag{2.22}$$

where neuron k is an output node. Differentiating Eq. 2.22 with respect to $y_j(n)$ we get

$$\frac{\partial \xi(n)}{\partial y_j(n)} = \sum_k e_k \frac{\partial e_k(n)}{\partial y_j(n)} \tag{2.23}$$

Next we use the chain rule to calculate the partial derivative $\partial e_k(n)/\partial y_j(n)$, and rewrite Eq. 2.23 in the equivalent form

$$\frac{\partial \xi(n)}{\partial y_j(n)} = \sum_k e_k \frac{\partial e_k(n)}{\partial v_k(n)} \frac{\partial v_k(n)}{\partial y_j(n)} \tag{2.24}$$

Note from Eqs. 2.8 and 2.7 that

$$\begin{aligned}e_k(n) &= d_k(n) - y_k(n) \\ &= d_k(n) - \varphi_k(v_k(n))\end{aligned} \tag{2.25}$$

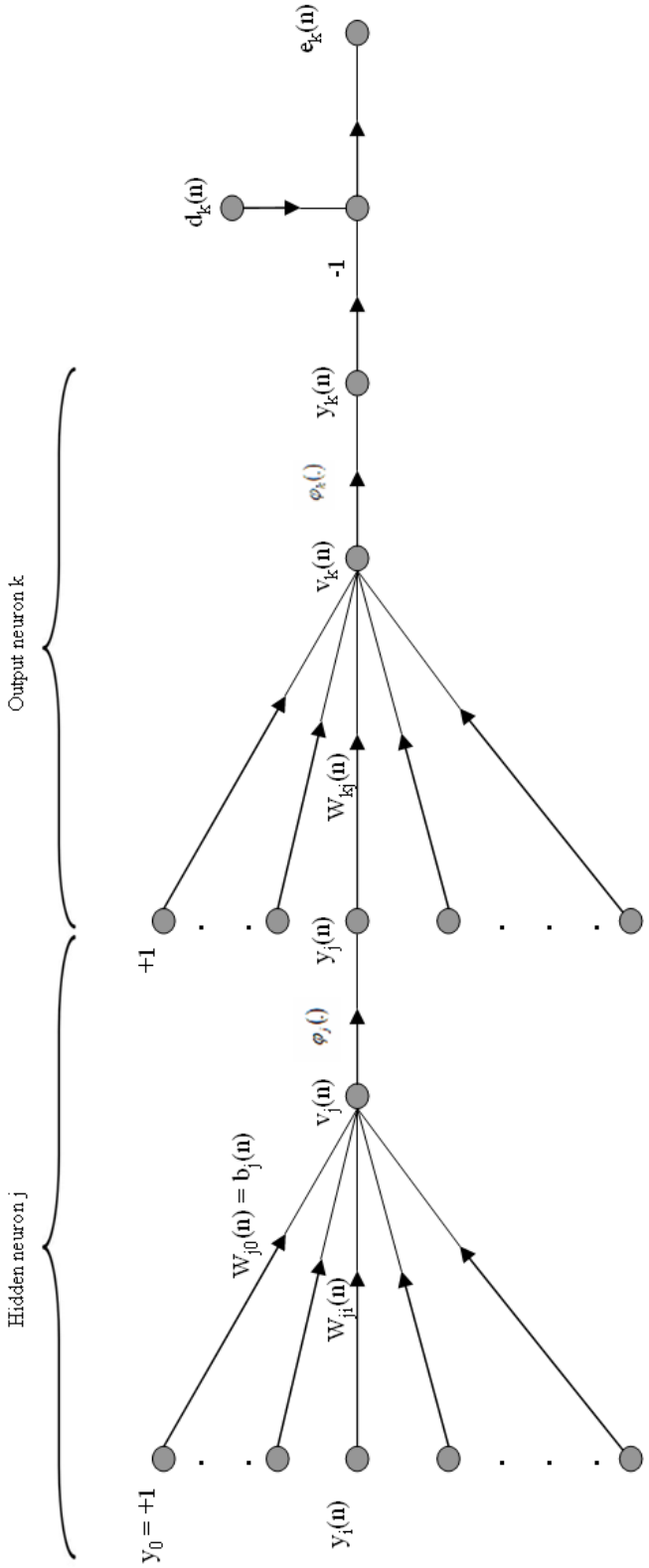


Figure 2.6 Signal-flow graph highlighting the details of output neuron k connected to hidden neuron j [1]

where index k is used in place of index j. Therefore

$$\frac{\partial e_k(n)}{\partial v_k(n)} = -\varphi'_k(v_k(n)) \tag{2.26}$$

And also note from Eq. 2.6 that

$$v_k(n) = \sum_{j=0}^{m} w_{kj}(n) y_j(n) \tag{2.27}$$

where $w_{kj}(n)$ denotes the synaptic weight connecting neuron j to neuron k at iteration n. Hence

$$\frac{\partial v_k(n)}{\partial y_j(n)} = w_{kj}(n) \tag{2.28}$$

By using Eqs. 2.26 and 2.28 in 2.24 we get the desired partial derivative

$$\begin{aligned} \frac{\partial \xi(n)}{\partial y_j(n)} &= -\sum_k e_k(n) \varphi'_k(v_k(n)) w_{kj}(n) \\ &= -\sum_k \delta_k(n) w_{kj}(n) \end{aligned} \tag{2.29}$$

where in the second line we have used the definition of the local gradient $\delta_k(n)$ given in Eq. 2.19 with the index substituted for j. Finally, using Eq. 2.29 in Eq. 2.20, we get the back-propagation formula for the local gradient $\delta_j(n)$ for a hidden neuron as described:

$$\delta_j(n) = \varphi'_j(v_j(n)) \sum_k \delta_k(n) w_{kj}(n) \tag{2.30}$$

2.2.4 Activation Functions

We see from Eqs. 2.19 and 2.30 that knowledge of the derivative of the activation function $\varphi(.)$ is necessary to calculate the local gradient $\delta_j(n)$ for both neurons in the output and hidden layers. For this derivative to exist $\varphi(.)$ must be continuously differentiable. A nonlinear activation function meeting this requirement, which is commonly used in MLPs is the *sigmoid* nonlinearity. Two forms of sigmoid activation function are described below.

2.2.4.1 Logistic Function

$$\varphi_j(v_j(n)) = \frac{1}{1 + \exp(-a v_j(n))} \qquad a > 0 \ and \ -\infty < v_j(n) < \infty \tag{2.31}$$

where a is the slope parameter of the sigmoid function. According to this nonlinearity the network output $y_j(n)$ lies in the interval $[0, 1]$.

2.2.4.2 Hyperbolic Tangent Function

$$\varphi_j(v_j(n)) = a \tanh(bv_j(n)) \qquad a, b > 0 \tag{2.32}$$

where a and b are constants. The hyperbolic tangent function is actually just the logistic function rescaled and biased. According to this nonlinearity the network output $y_j(n)$ has the range $[-1, +1]$.

2.2.5 Momentum Term

Gradient descent can progress very slow if the learning rate parameter η is small resulting in a very slow rate of convergence or it can oscillate if η is too large resulting in divergence. These situations can be remedied by introducing a *momentum term* to the delta rule of Eq. 2.18 as shown by

$$\Delta w_{ji}(n) = \alpha \Delta w_{ji}(n-1) + \eta \delta_j(n) y_i(n) \qquad 0 \leq \alpha < 1 \tag{2.33}$$

where α is the momentum term. Eq. 2.33 is called *generalized delta rule* which includes the delta rule of Eq. 2.18 as a special case for $\alpha = 0$. Generalized delta rule allows calculation of $\Delta w_{ji}(n)$ in a recursive manner. Introduction of the momentum term also prevents the back-propagation algorithm from terminating at a local minimum.

CHAPTER 3. LITERATURE REVIEW

3.1 Neural Networks in Material Science

After Rumelhart and his coworkers' pioneering publication in Nature [15] presenting "back-propagation" as a means of training multilayer perceptrons (multilayer feed-forward neural networks) more researchers from various fields began to employ neural networks to solve complicated nonlinear problems in science and engineering.

Joyce et al. [16] used back-propagation neural networks to predict glass transition temperature of linear homopolymers from their monomer structures alone. Cundari et al. [17] compared neural network models to quantum mechanical models for predicting molecular properties of inorganic systems and concluded that neural networks give more accurate predictions. Asada et al. [18] used a back-propagation feed-forward network to predict superconducting transition temperature of Ca doped $YBa_2Cu_3O_z$ as a function of chemical composition.

Malinov et al. used artificial neural networks for modeling TTT diagrams for $\beta \rightarrow \alpha + \beta$ phase transformation in Ti alloys [19], the relationship between processing and working conditions and mechanical properties of Ti alloys [20], and γ-Ti aluminides [21], fatigue stress life diagrams of the Ti-6Al-4V alloy [22] and corrosion resistance of Ti alloys [23]. Parametric studies were conducted to investigate the influence of various input parameters on the material properties and inverse models were developed to optimize input parameters to obtain desirable material properties. Performance of various learning algorithms were investigated and Bayesian regularization in combination with Levenberg-Marquardt algorithm was found to give the best results.

MacKay [24] developed a Bayesian framework for back-propagation feed-forward networks, which enables calculation of error bars for model predictions and quantifies the significance of

each input parameter automatically. This approach has advantages in analyzing the influence of many input parameters on a specific output, however the algorithm is complex and slows down network convergence.

Gavard et al. [25] studied formation of austenite during continuous heating of steels using neural networks with a Bayesian framework. Austenite start and finish temperatures are predicted as a function of chemical composition and heating rate. Vermeulen at al. demonstrated that a feed-forward neural network can estimate martensite start temperature [26] as a function of chemical composition in vanadium containing steels and Jominy hardness profiles [27] of steels as a function of chemical composition and austenitizing temperature. In a later study the author and his coworkers [28] predicted continuous cooling transformation diagrams of some selected vanadium steels as a function of chemical composition, austenitizing temperature and cooling rate using artificial neural networks.

Bhadeshia et al. [29] used a Bayesian framework to analyze the influence of welding process, alloying elements, microstructure and test temperature on the Charpy toughness of steel welds. Ichikawa and coworkers [30] used a classification neural network based on a Bayesian framework to predict occurrence of solidification cracking in low alloy steel welds. Cool et al. [31] used a committee of neural networks to predict strength of steel welds. This approach was found to improve quality of prediction in the regions of the input space where data is sparse and reliability is low.

Yoshitake et al. [32] modeled temperature dependence of lattice constants of γ and γ' phases in Ni based superalloys using a committee of neural networks within a Bayesian framework. Fujii et al. [33] used the same approach to predict the fatigue crack growth rate in Ni-based superalloys as a function of 51 variables. Neural networks were shown to enable studying the effects of individual input parameters on the outputs in isolation in cases where variables intrinsically depend on each other.

In a study of the fatigue thresholds in nickel-based superalloys, Schooling et al. [34] compared a "neurofuzzy" modeling approach with a Bayesian neural network. The application of fuzzy rules to the network involves biasing the inputs according to human experience. Neu-

rofuzzy modeling was found to be advantageous where the training data is limited and the relationship between the inputs and the outputs is rather simple.

Singh et al. [35] used a Bayesian neural network to predict the yield and tensile strength of rolled steel sheets as a function of chemical composition and processing parameters. Korczak et al. [36] used a back-propagation feed-forward network to model distribution of ferritic grain size and mechanical properties along the thickness of hot rolled steel plates as a function of chemical composition, microstructure and processing parameters. Vermeulen et al. [37] also used a back-propagation feed-forward neural network to predict the finishing temperature on a rolling mill as a function of processing parameters. Larkiola et al. [38] used back-propagation feed-forward neural networks to predict deformation resistance of steel, and the friction parameter in cold rolling as a function of alloy composition and processing parameters. Liu et al. [39] modeled the relationship between microstructure and mechanical properties of hot rolled steel strips using neural networks.

Bhadeshia [40] reviewed the use of neural networks in material science and included notes on how theory of neural networks can be put into practice with suggestions on how performance of neural networks can be improved where necessary.

Artificial neural networks were also used frequently in composites design to model non-linear material behavior. Roberts et al. [41] used artificial neural networks to predict damage evolution in forged aluminum matrix composites as a function of forging parameters. Rai and Pithumani [42] modeled cure processes of polymer matrix composites using feed-forward neural networks while Su et al. [43] used a recurrent neural network to predict the degree of cure in polymer matrix composites. Recurrent neural networks are used to model dynamic nonlinear relationships between input and output parameters.

Al-Assaf and El-Kadi [44] applied multilayer feed-forward neural networks to predict fatigue life of unidirectional composites. In a later study, the authors investigated the performance of different kinds of neural networks such as modular, self-organizing, radial basis, and principal component analysis networks using the same database. It was concluded that modular neural networks gave better fatigue life predictions than feed-forward neural networks. [45]

Zhang et al. used multilayer feed-forward neural networks trained with the back-propagation algorithm to predict specific wear rate and friction coefficient of short fiber reinforced composites. The influence of learning algorithms, ANN structure and number of training examples on the performance of neural networks were analyzed in detail. In a later study [46] the author and his coworkers used neural networks to predict the erosive wear rate of a few polymers as a function of testing conditions and material properties. Systematic parametric studies were conducted to investigate which characteristic property has the the most pronounced influence on the erosive wear rate. [47]

Zhang et al. employed neural networks to predict storage and loss modului of short fiber reinforced composites as a function of material composition and temperature. Bayesian regularization of back-propagation algorithm was used to determine the optimum number of neurons in the hidden layers. It was concluded that one output networks lead to better predictive performance in cases where training data is limited and larger training data sets are required for more complex nonlinear input-output relationships. [48]

Zhang and Friedrich [49] reviewed the use of neural networks in polymer composites modeling and studied the influence of input parameters, number of neurons in hidden layers and the size of training set on neural network performance by giving examples from the cited references. They also pointed out right and wrong practices of neural networks and gave suggestions on improving the models.

3.2 Developing Mechanical Models Using Neural Networks

Artificial neural networks were frequently employed in modeling of constitutive behaviour of monolithic and composite materials often along with finite element modeling. Rao and Mukherjee [50] used artificial neural networks to model the effect of interfacial shear strength and fiber volume fraction on the stress-strain relationship and the extent of debonding of ceramic matrix composites. Haj-Ali et al. [51] employed neural networks to develop constitutive models for unidirectional composites.

It is known that using additional input parameters derived from the existing inputs for the

known nonlinear input-output relationships improves neural network convergence. Mukherjee and coworkers [52] studied the effect of fiber volume fraction and loading direction on the strain hardening behavior of metal matrix composites by using neural networks and confirmed that using squares of the fiber volume fraction and the matrix hardening exponent as additional input parameters leads to faster convergence and a marginally lower mean squared error. In another study Rao et al. [53] used a multilayer feed-forward neural network trained with the back-propagation algorithm to model the effect of interfacial shear strength on the stress-strain relationship of whisker reinforced ceramic matrix composites. They used the interfacial shear strength and its square as an additional input parameter, and a genetic algorithm to assign the initial weights of the network to speed up the network convergence.

In a recent study Haj-Ali and Kim used feed-forward neural networks trained with an adaptive back-propagation algorithm, which adds neurons to hidden layers at specified intervals depending on network performance, to develop multi-axial constitutive models for fiber reinforced composites. Separate neural networks were trained to model inelastic and total strains. Moreover the effect of using vector and scalar outputs were investigated, however similar results were obtained. Inverse neural network models were developed and integrated with FEA to validate network performance. Both forward and inverse models were trained and tested with experimental data. [54]

Huber et al. studied determination of constitutive properties for a fictious material from spherical indentation tests using neural networks for the case of plasticity with pure kinematic hardening [55], and plasticity with isotropic and kinematic hardening [56]. Inverse models were developed using neural networks to relate depth-load response to constitutive equation parameters. Huber and his coworkers [57, 58] also used neural networks to predict Poisson's ratio from spherical nanoindentation tests.

Muliana et al. [59] employed neural networks to model load displacement relationship of annealed copper obtained from the loading portion of nanoindentation tests. The results from FEA were validated by experimental data and used to train and test neural networks with an adaptive back-propagation algorithm. The trained ANNs were used to solve the inverse

problem of extracting the plastic flow properties from the load-displacement behavior. Huber et al. also used the information from both loading and unloading portions of the indentation curve to predict stress-strain behavior of thin metallic films on substrates form pyramidal [60] and spherical [61] indentation tests.

The present study is the first application of ANN in the analysis and interpretation of engineering diffraction data. It demonstrates the power of ANN in conducting robust inverse analysis of such data. The approach developed and presented here can also be employed in the optimization of engineering diffraction experiments to increase their accuracy and efficiency.

CHAPTER 4. SENSITIVITY STUDY

4.1 Introduction

Before developing neural network models it is necessary to conduct a sensitivity study to analyze the influence of each parameter on the stress-strain response of the material upon loading and unloading. In order to analyze the influence of each variable individually six of the variables are set to their mean value while the analyzed variable is changed between their minimum and maximum at five points. Five points are found to be sufficient to analyze the parametric sensitivity of the constitutive behavior of the composites. The loading-unloading curves obtained are plotted in series for fiber elastic strain and composite total stain as is done throughout this thesis to make it easier to compare with experimental data.

4.2 Influence of σ_0^W

Table 4.1 Change in σ_0^W for the 20% and 40% composites

#	σ_0^W (MPa)	σ_1 (MPa)	θ_0 (MPa)	θ_1 (MPa)	σ_0^{BMG} (MPa)	n	T (°C)
1	1000	700	750000	1320	2000	8	300
2	1125	700	750000	1320	2000	8	300
3	1250	700	750000	1320	2000	8	300
4	1375	700	750000	1320	2000	8	300
5	1500	700	750000	1320	2000	8	300

σ_0^W denotes the yield strength of the tungsten fibers. (Figure 1.1) Tables 4.1 and 4.2 show the values of the seven parameters at each iteration to analyze the influence of σ_0^W on the constitutive behavior of the 20% and 40%, and 60% and 80% composites respectively. The FEM does not converge for the 60% and 80% composites when σ_0^W is set to 1000 MPa therefore,

Table 4.2 Change in σ_0^W for the 60% and 80% composites

#	σ_0^W **(MPa)**	σ_1 **(MPa)**	θ_0 **(MPa)**	θ_1 **(MPa)**	σ_0^{BMG} **(MPa)**	**n**	**T (°C)**
1	1125	700	750000	1320	2000	8	300
2	1175	700	750000	1320	2000	8	300
3	1250	700	750000	1320	2000	8	300
4	1375	700	750000	1320	2000	8	300
5	1500	700	750000	1320	2000	8	300

the sensitivity of the variables are analyzed within a different range for those composites.

Figures 4.1 through 4.8 clearly show that σ_0^W has a major influence on the constitutive behavior of the composites. σ_0^W has a stronger effect on the composite total strain than it has on the fiber elastic strain. Moreover, the effect of σ_0^W on the composite total strain becomes incrementally less influential. For instance changing the yield strength of the fibers from 1350 to 1500 MPa has a relatively minor effect on the composite total strain when compared to the other iterations for the 20% composite. This effect becomes more pronounced with increasing fiber volume fraction.

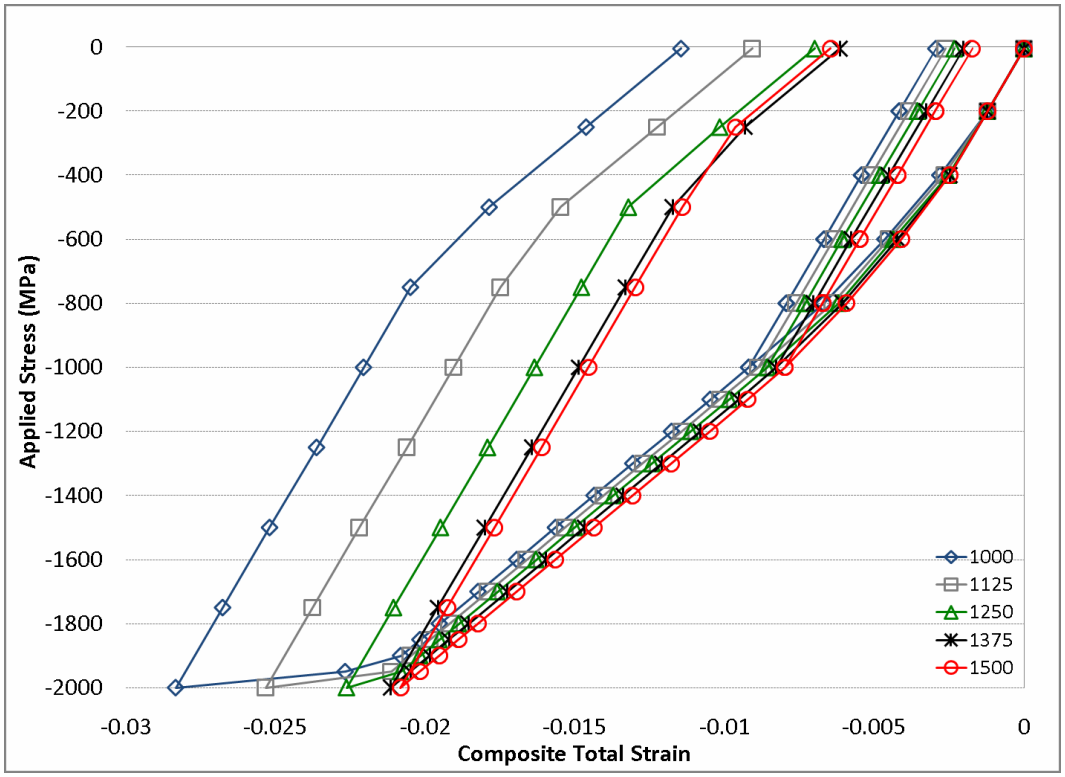


Figure 4.1 Influence of σ_0^W on the composite total strain for the 20% composite as predicted by FEM

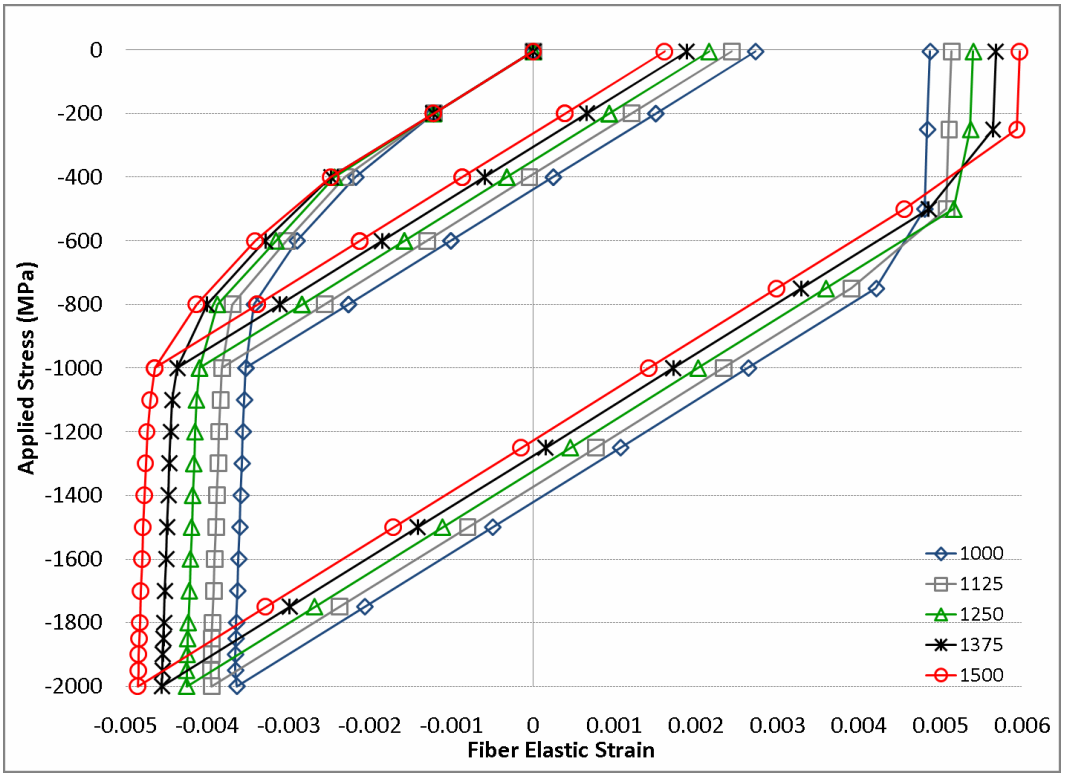


Figure 4.2 Influence of σ_0^W on the fiber elastic strain for the 20% composite as predicted by FEM

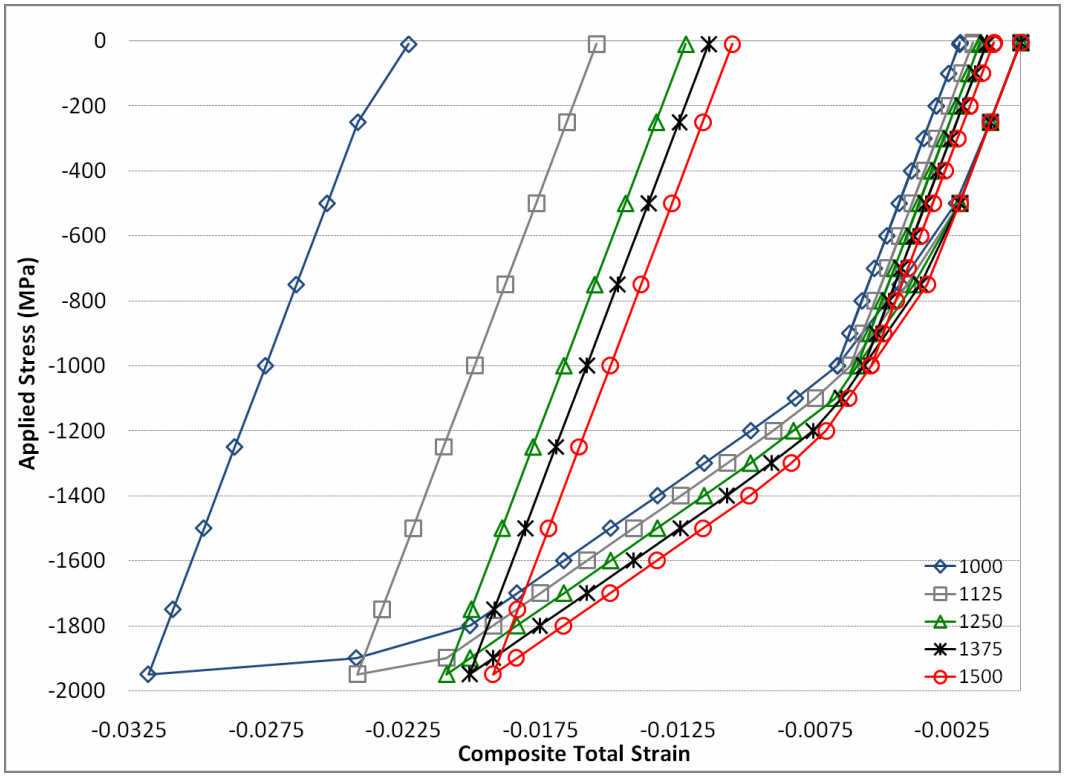


Figure 4.3 Influence of σ_0^W on the composite total strain for the 40% composite as predicted by FEM

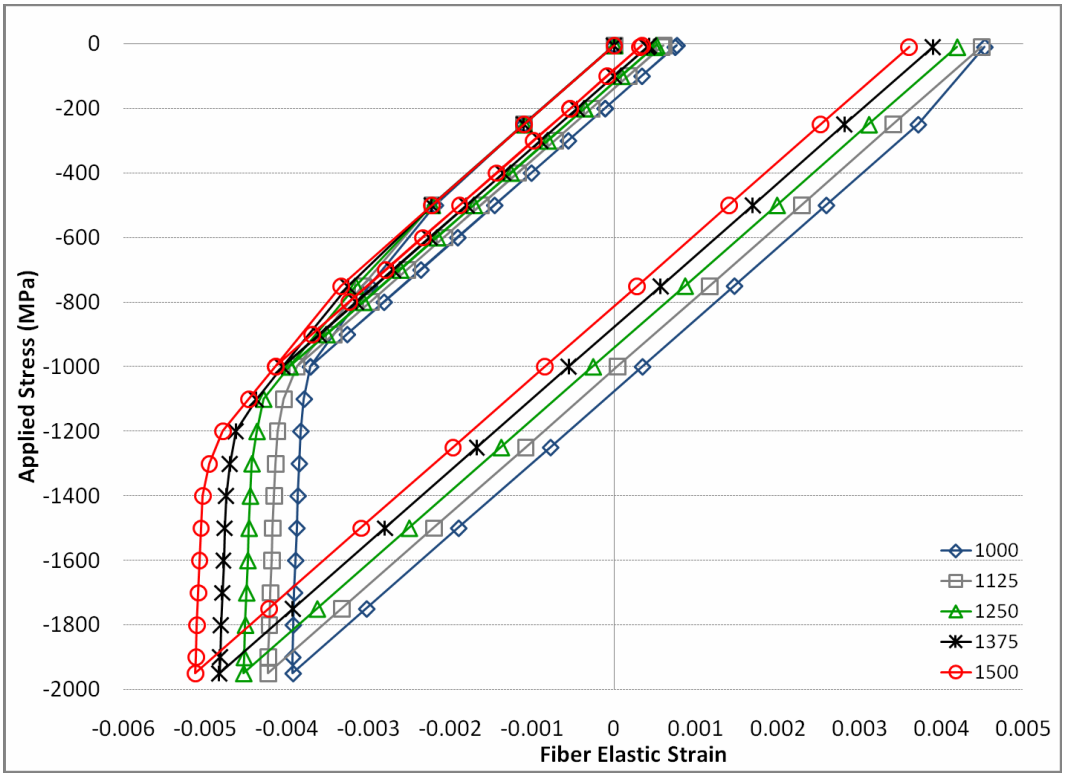


Figure 4.4 Influence of σ_0^W on the fiber elastic strain for the 40% composite as predicted by FEM

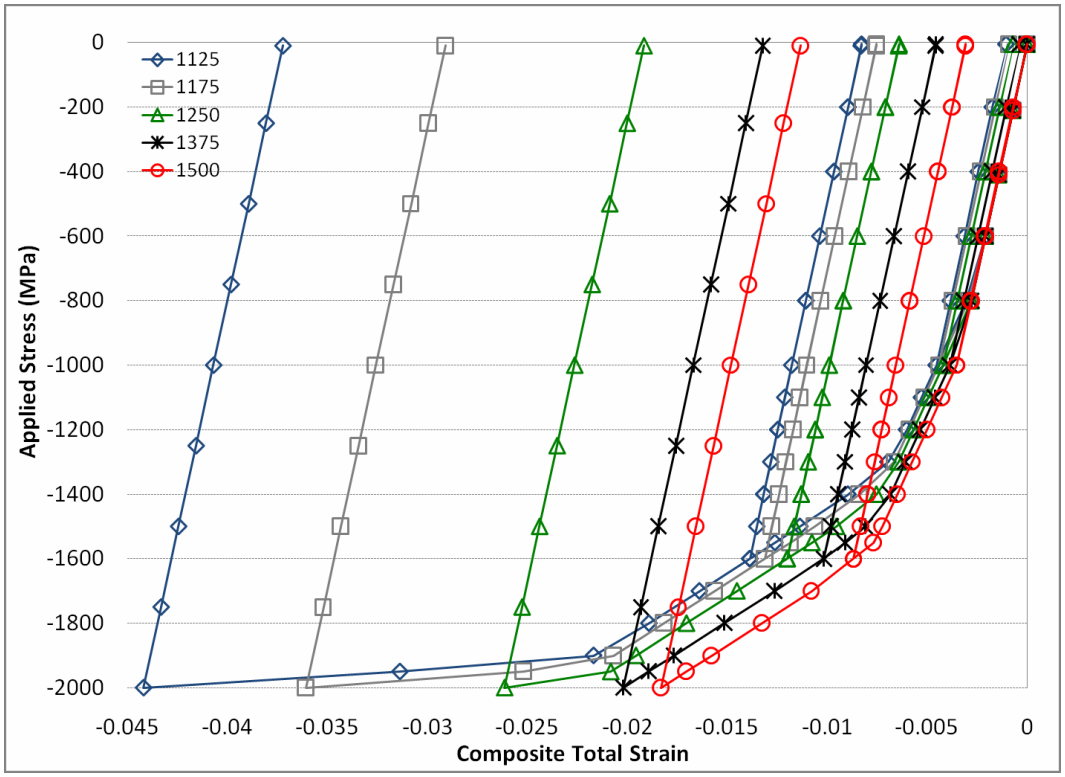


Figure 4.5 Influence of σ_0^W on the composite total strain for the 60% composite as predicted by FEM

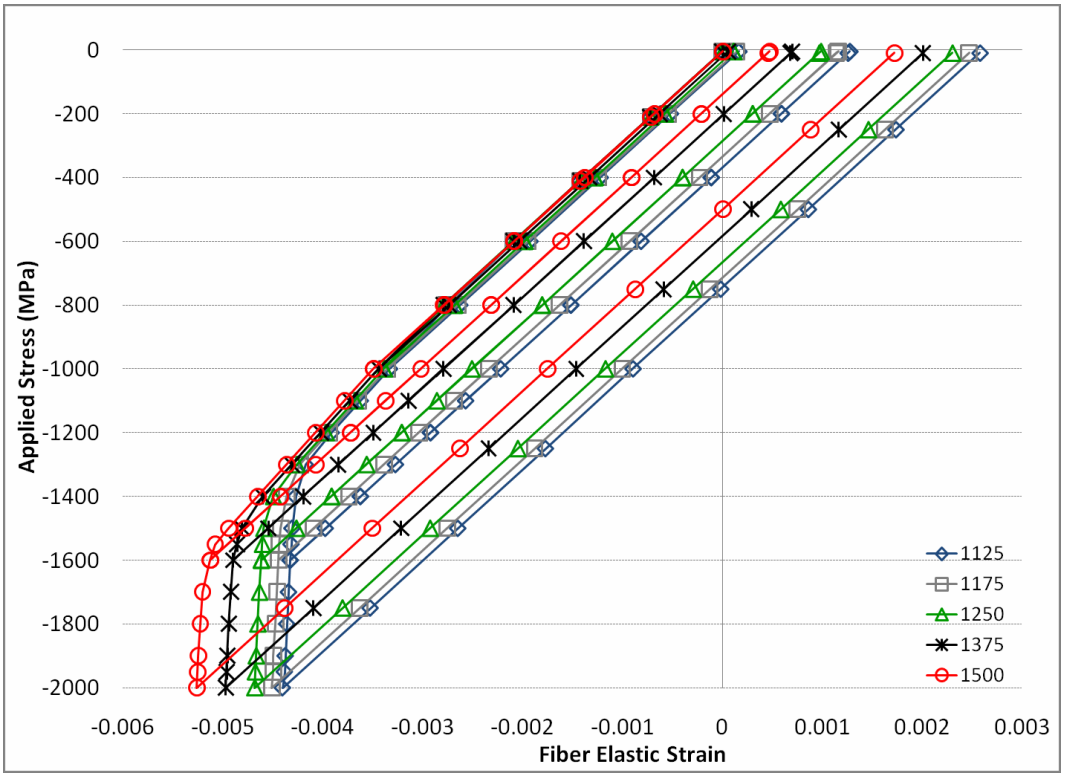


Figure 4.6 Influence of σ_0^W on the fiber elastic strain for the 60% composite as predicted by FEM

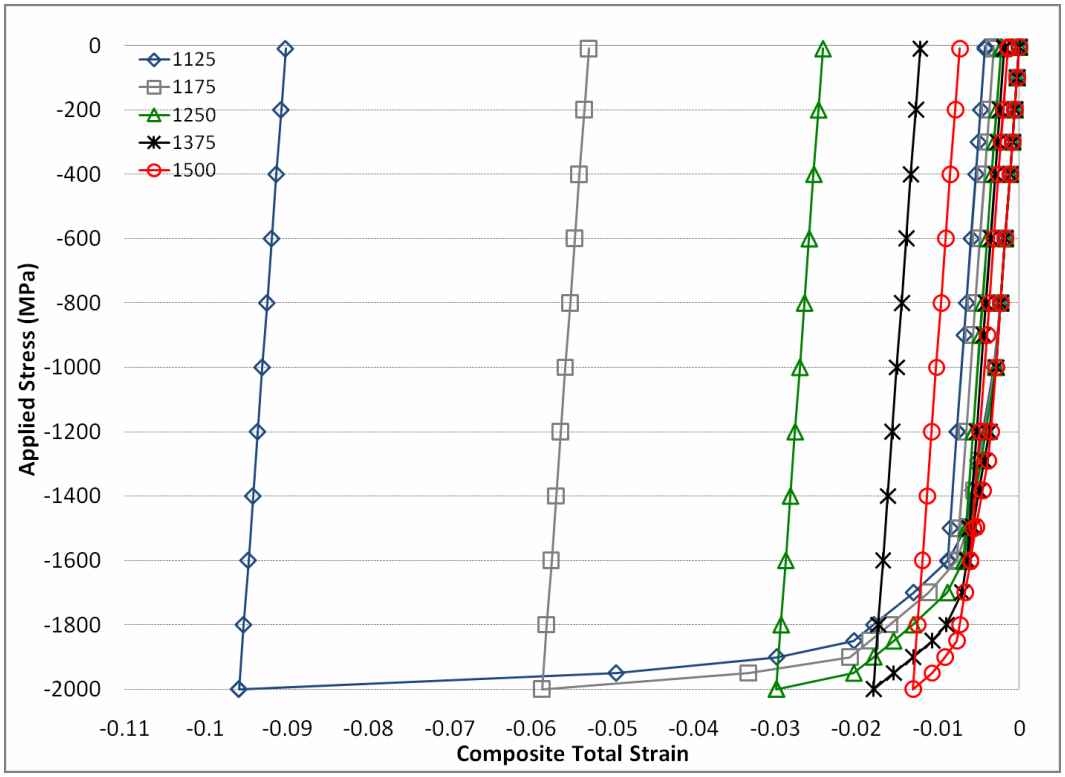


Figure 4.7 Influence of σ_0^W on the composite total strain for the 80% composite as predicted by FEM

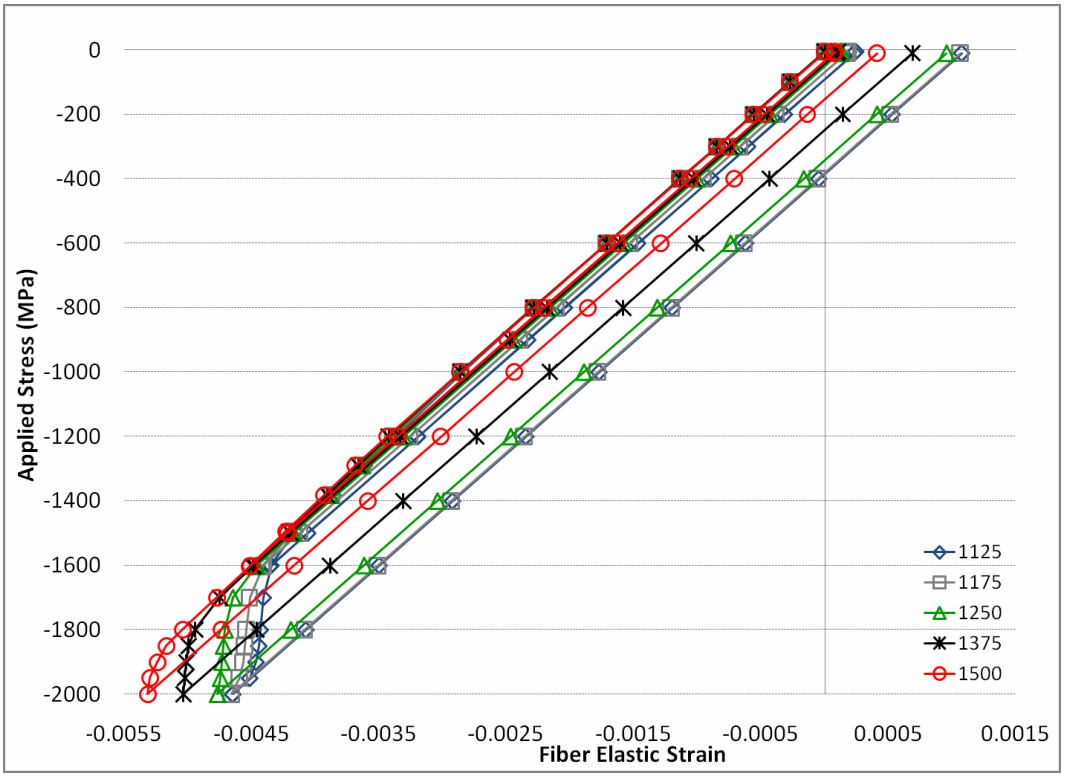


Figure 4.8 Influence of σ_0^W on the fiber elastic strain for the 80% composite as predicted by FEM

4.3 Influence of σ_1

σ_1 denotes the initial hardening of the fibers. (Figure 1.1) Tables 4.3 and 4.4 show the values of the seven parameters at each iteration to analyze the effect of σ_1 on the constitutive behavior of the composites. The FEM does not converge for the 80% composite when σ_1 is set to 500 MPa. Therefore, the effect of σ_1 is analyzed within a different range for that particular case.

Figures 4.9 through 4.16 show that σ_1 has a major influence on the constitutive behavior of the composites. The effect of σ_1 on the fiber elastic strain is rather uniform whereas it becomes less influential with successive stress increments for composite total strain. This effect becomes more pronounced with increasing fiber volume fraction.

Table 4.3 Change in σ_1 for the 20%, 40% and 60% composites

#	σ_0^W **(MPa)**	σ_1 **(MPa)**	θ_0 **(MPa)**	θ_1 **(MPa)**	σ_0^{BMG} **(MPa)**	**n**	**T (°C)**
1	1250	500	750000	1320	2000	8	300
2	1250	600	750000	1320	2000	8	300
3	1250	700	750000	1320	2000	8	300
4	1250	800	750000	1320	2000	8	300
5	1250	900	750000	1320	2000	8	300

Table 4.4 Change in σ_1 for the 80% composite

#	σ_0^W **(MPa)**	σ_1 **(MPa)**	θ_0 **(MPa)**	θ_1 **(MPa)**	σ_0^{BMG} **(MPa)**	**n**	**T (°C)**
1	1250	600	750000	1320	2000	8	300
2	1250	650	750000	1320	2000	8	300
3	1250	700	750000	1320	2000	8	300
4	1250	800	750000	1320	2000	8	300
5	1250	900	750000	1320	2000	8	300

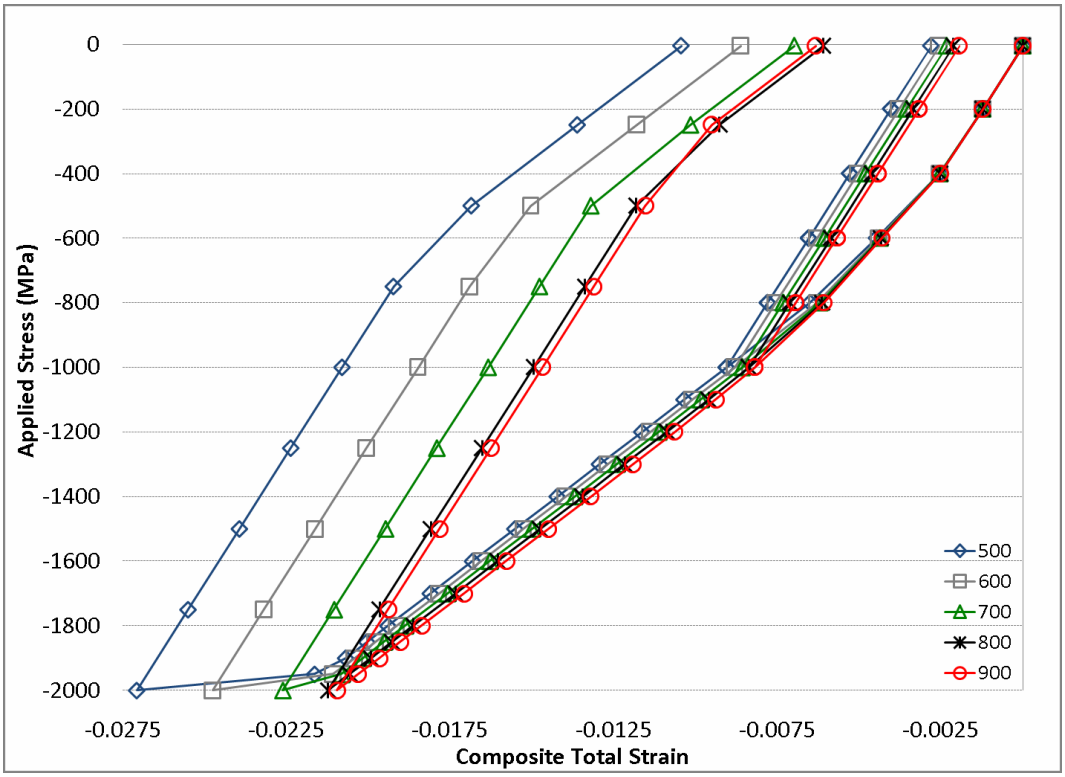


Figure 4.9 Influence of σ_1 on the composite total strain for the 20% composite as predicted by FEM

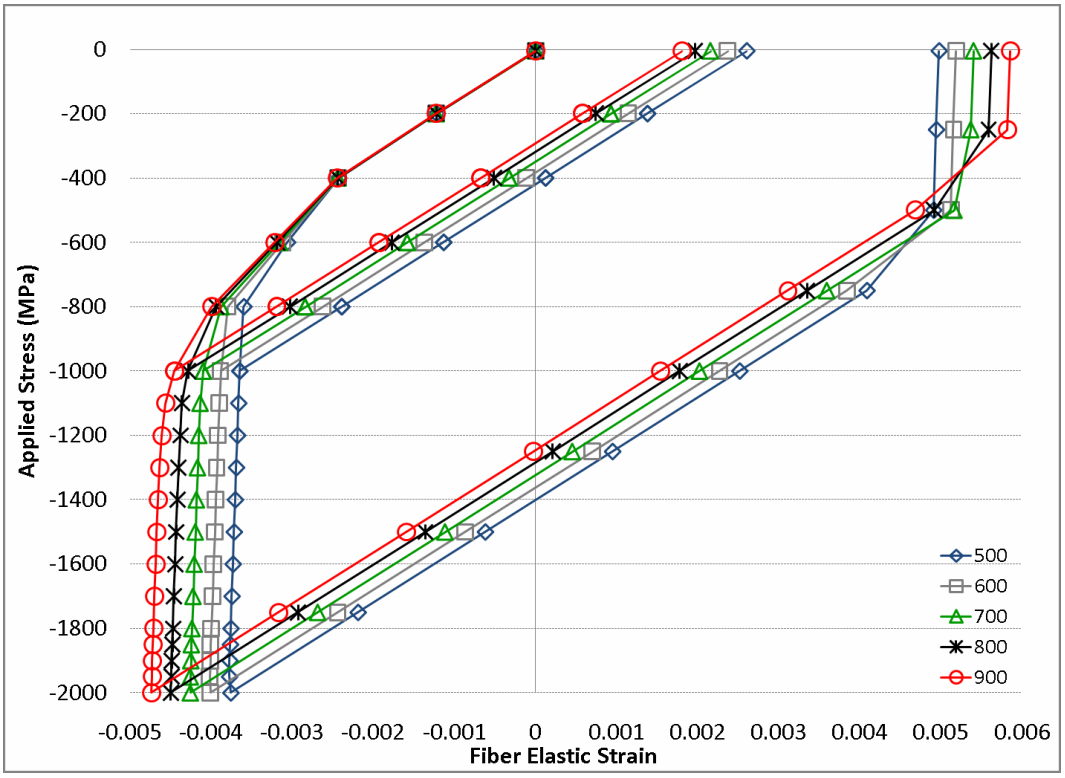


Figure 4.10 Influence of σ_1 on the fiber elastic strain for the 20% composite as predicted by FEM

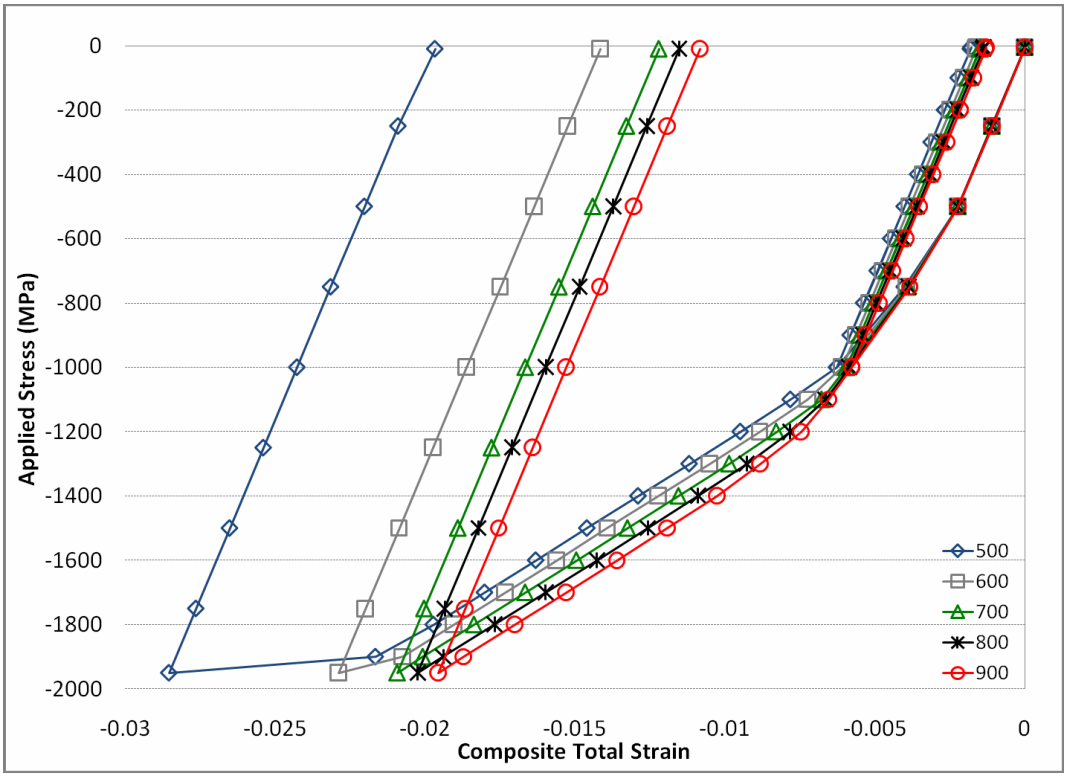


Figure 4.11 Influence of σ_1 on the composite total strain for the 40% composite as predicted by FEM

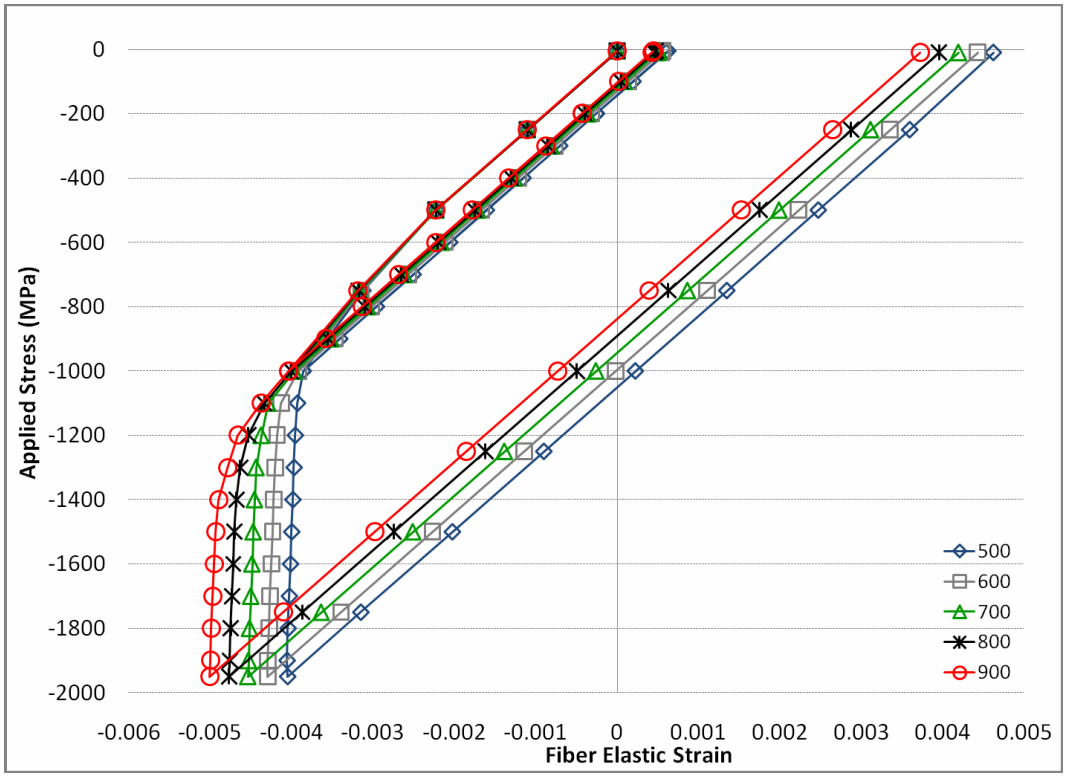


Figure 4.12 Influence of σ_1 on the fiber elastic strain for the 40% composite as predicted by FEM

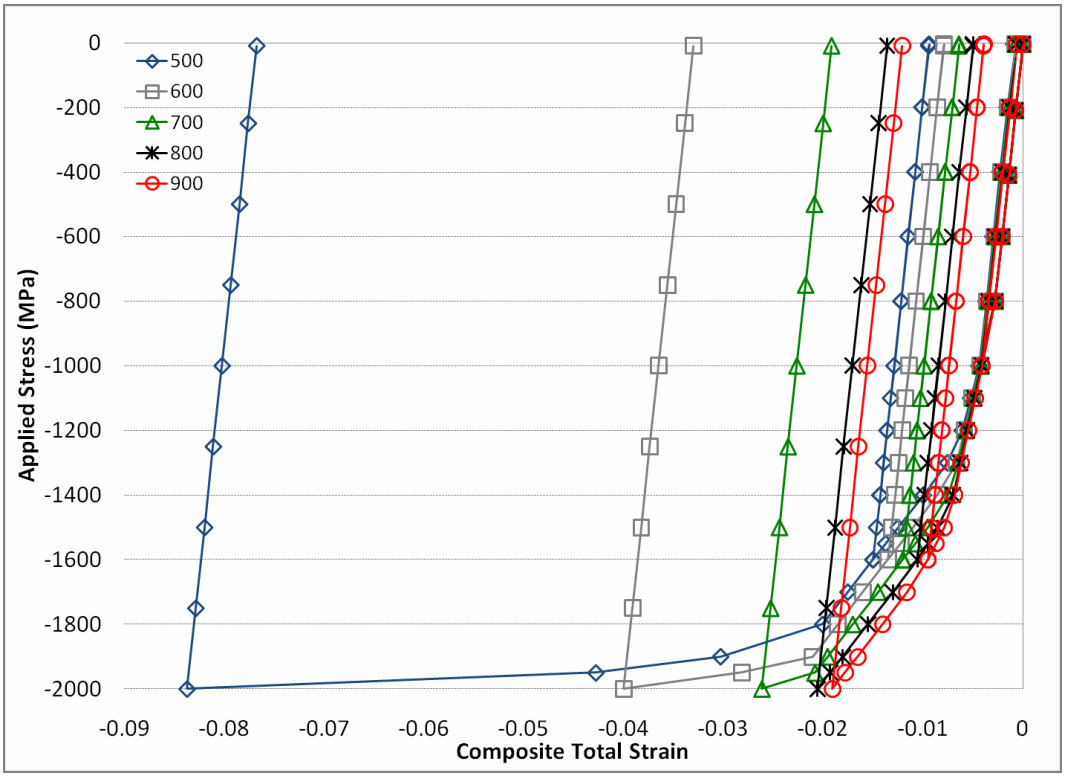


Figure 4.13 Influence of σ_1 on the composite total strain for the 60% composite as predicted by FEM

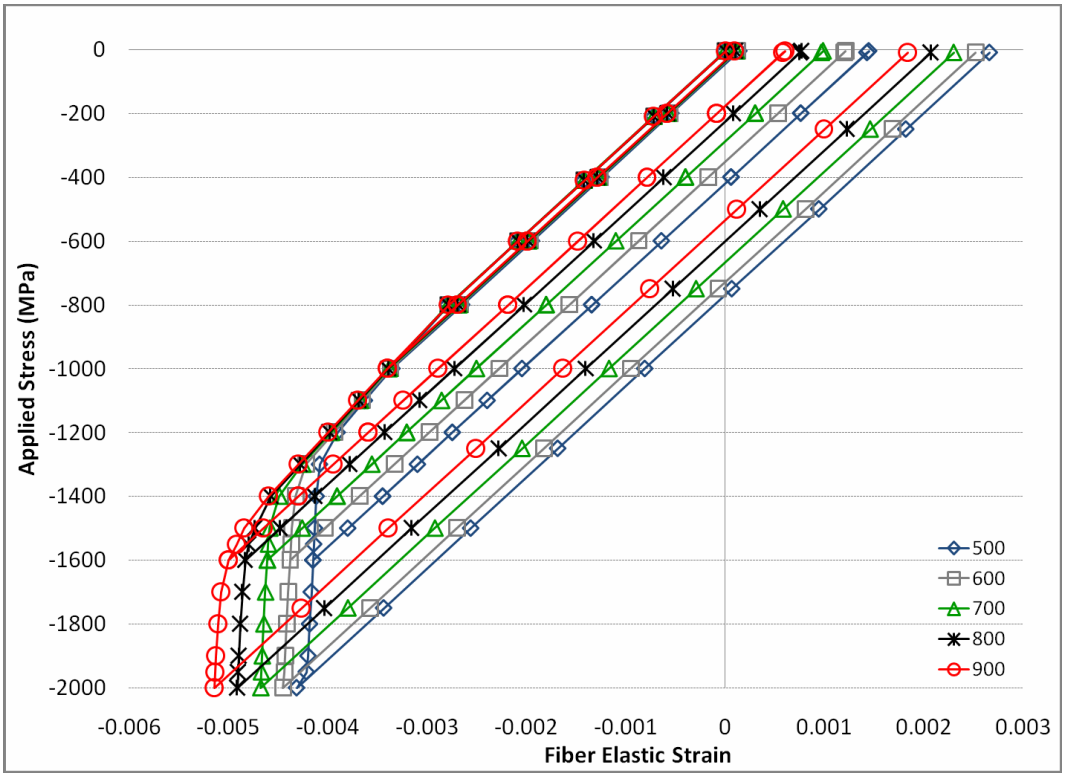


Figure 4.14 Influence of σ_1 on the fiber elastic strain for the 60% composite as predicted by FEM

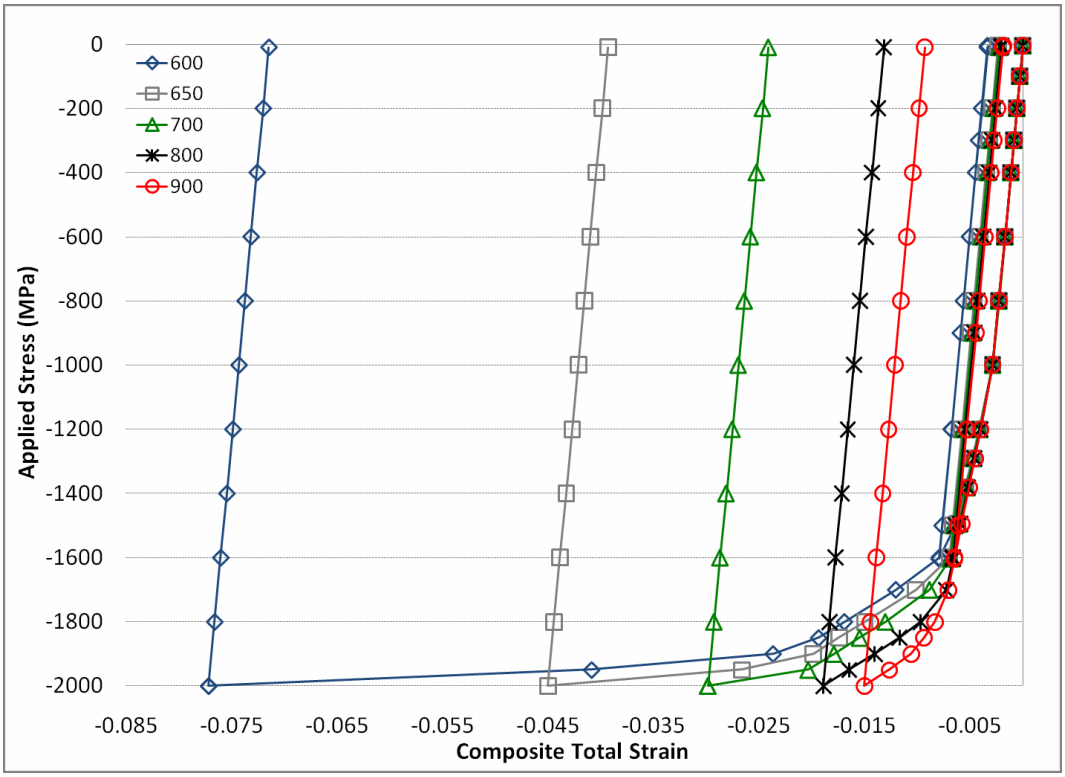


Figure 4.15 Influence of σ_1 on the composite total strain for the 80% composite as predicted by FEM

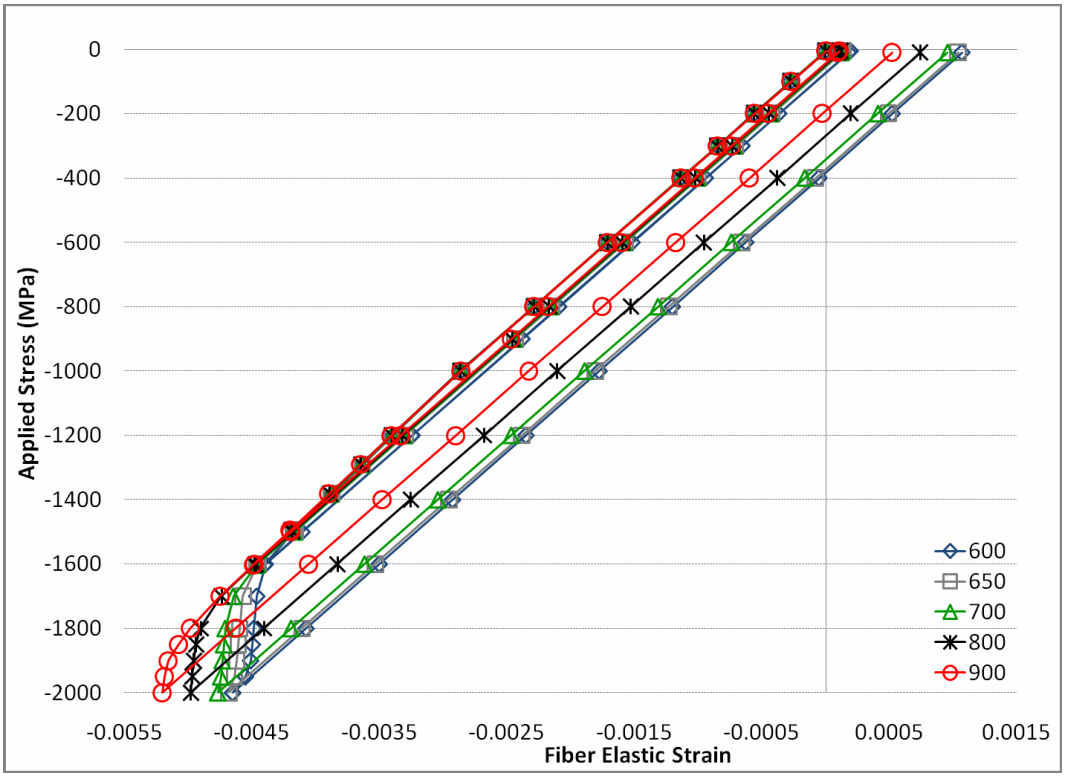


Figure 4.16 Influence of σ_1 on the fiber elastic strain for the 80% composite as predicted by FEM

4.4 Influence of θ_0

θ_0 denotes the slope of the stress-strain curve for the initial hardening of the fibers. (Figure 1.1) Table 4.5 shows the values of the seven parameters at each iteration to analyze the influence of θ_0 on the constitutive behavior of the composites.

Figures 4.17 through 4.24 show that θ_0 has a minor influence on the fiber elastic strain and the composite total strain during the inital yielding of the fibers and the composite. The effect of θ_0 becomes more pronounced with increasing fiber volume fraction. Particularly, changing θ_0 from 600000 to 675000 MPa has a noticable effect on the fiber elastic strain and composite total strain for the 80% composite.

Table 4.5 Change in θ_0

#	σ_0^W (MPa)	σ_1 (MPa)	θ_0 (MPa)	θ_1 (MPa)	σ_0^{BMG} (MPa)	n	T (°C)
1	1250	700	600000	1320	2000	8	300
2	1250	700	675000	1320	2000	8	300
3	1250	700	750000	1320	2000	8	300
4	1250	700	825000	1320	2000	8	300
5	1250	700	900000	1320	2000	8	300

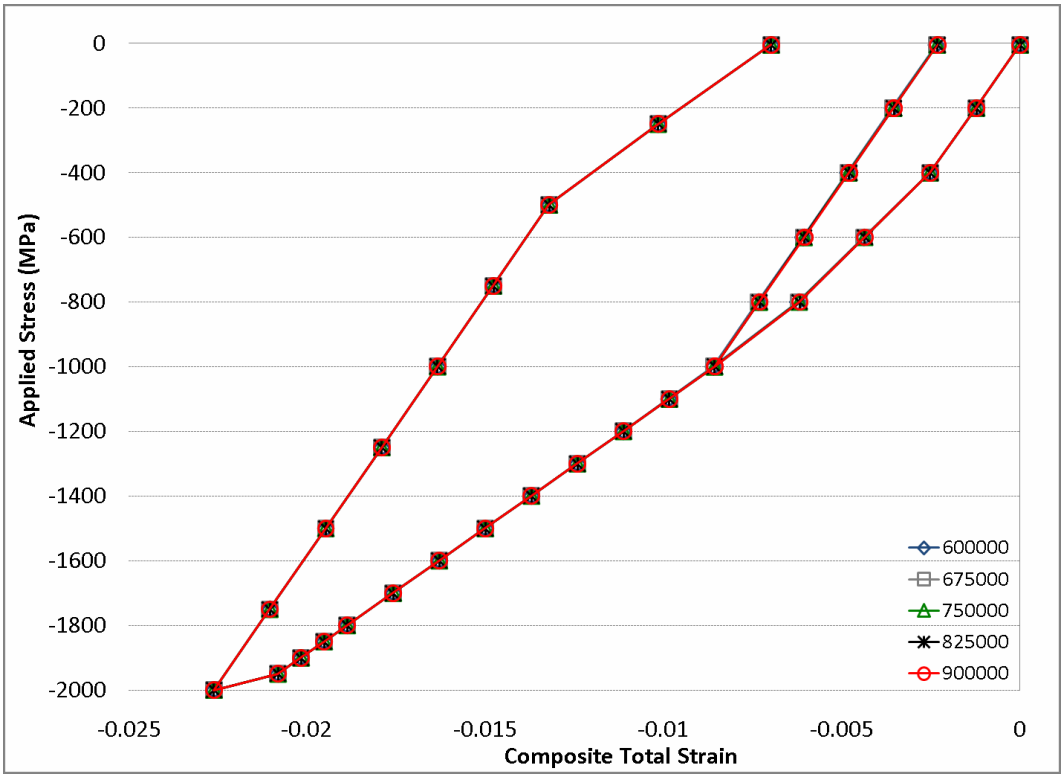


Figure 4.17 Influence of θ_0 on the composite total strain for the 20% composite as predicted by FEM

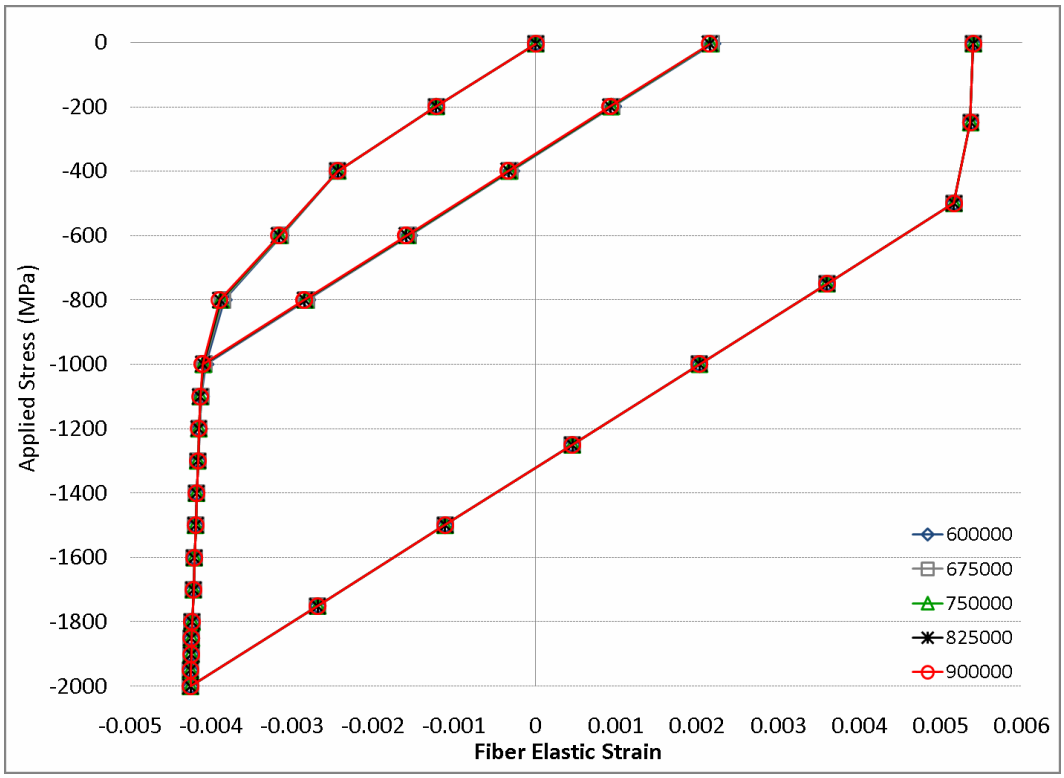


Figure 4.18 Influence of θ_0 on the fiber elastic strain for the 20% composite as predicted by FEM

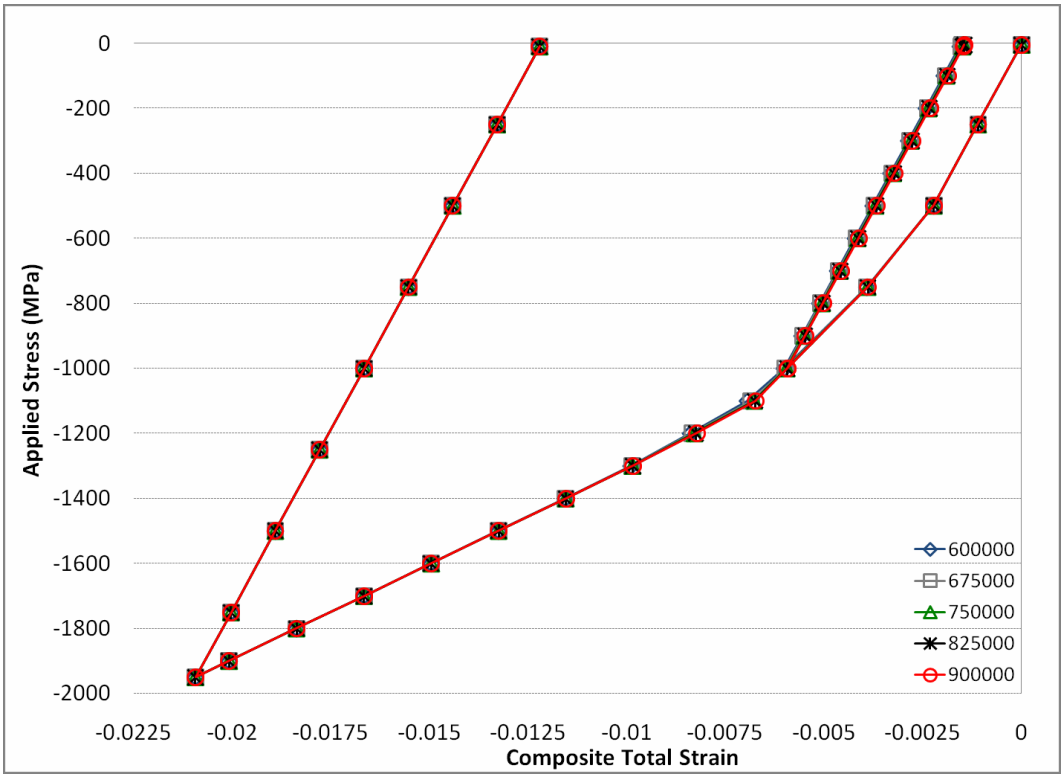


Figure 4.19 Influence of θ_0 on the composite total strain for the 40% composite as predicted by FEM

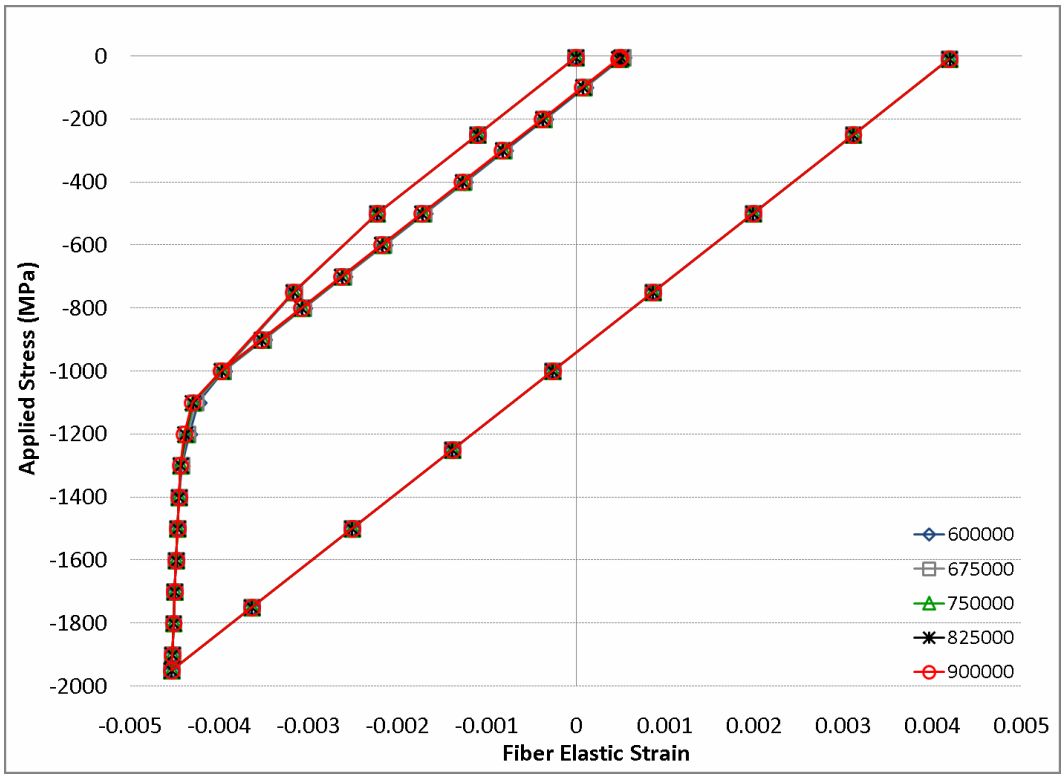


Figure 4.20 Influence of θ_0 on the fiber elastic strain for the 40% composite as predicted by FEM

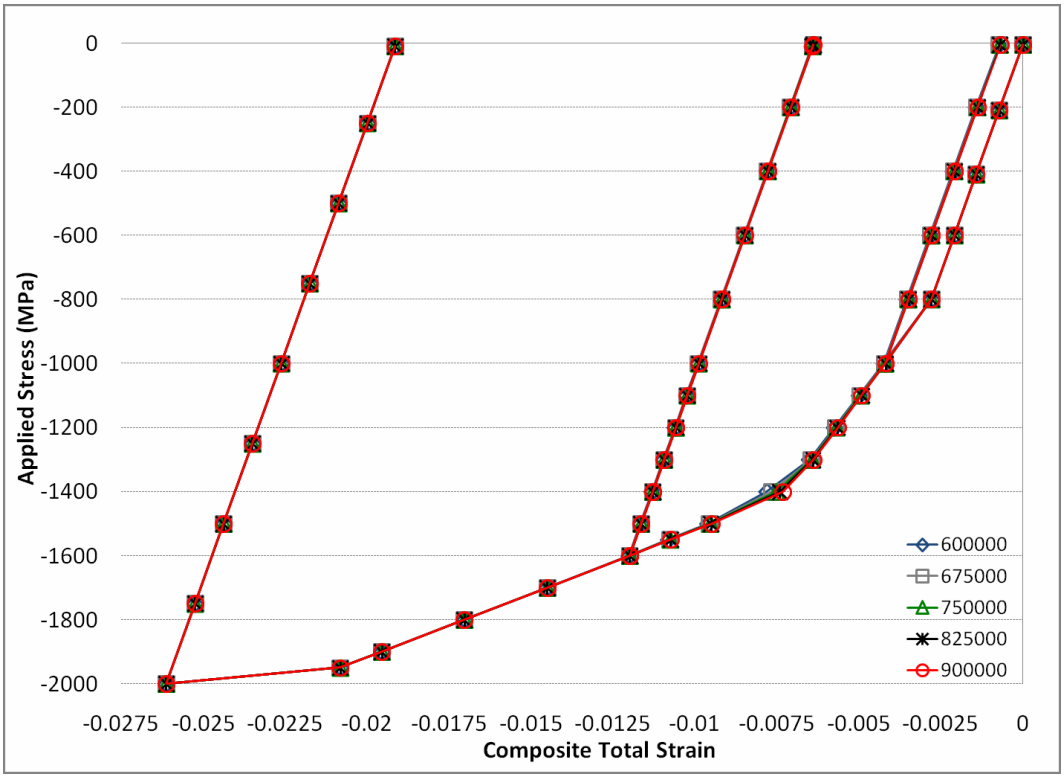


Figure 4.21 Influence of θ_0 on the composite total strain for the 60% composite as predicted by FEM

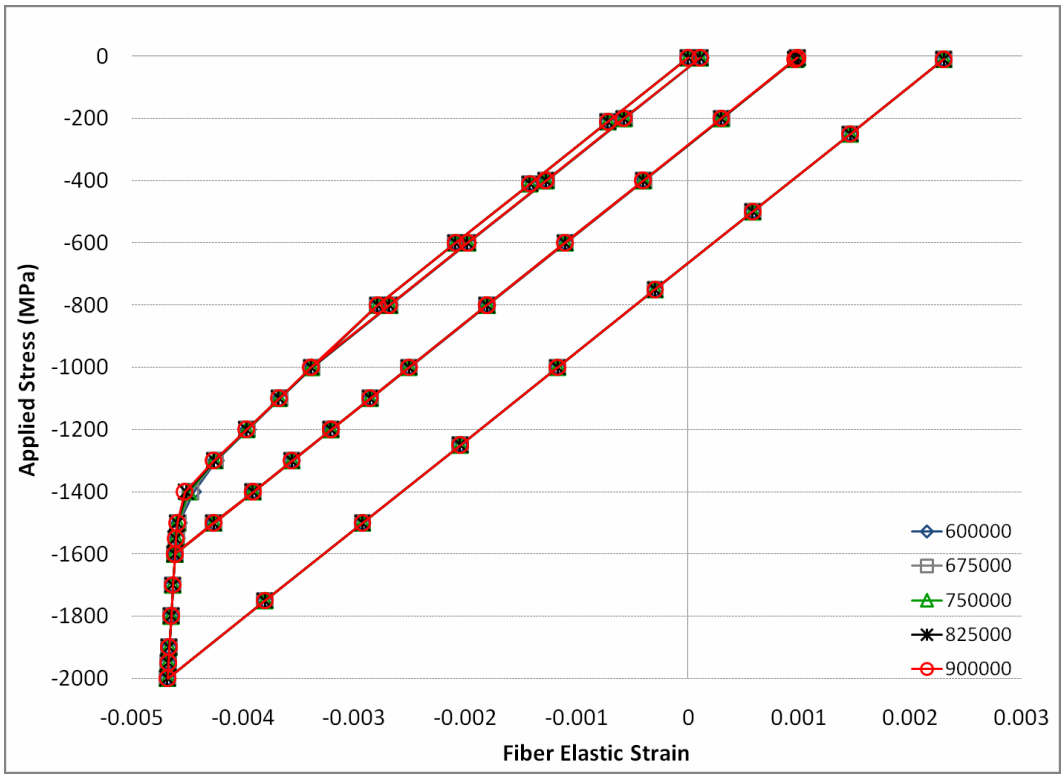


Figure 4.22 Influence of θ_0 on the fiber elastic strain for the 60% composite as predicted by FEM

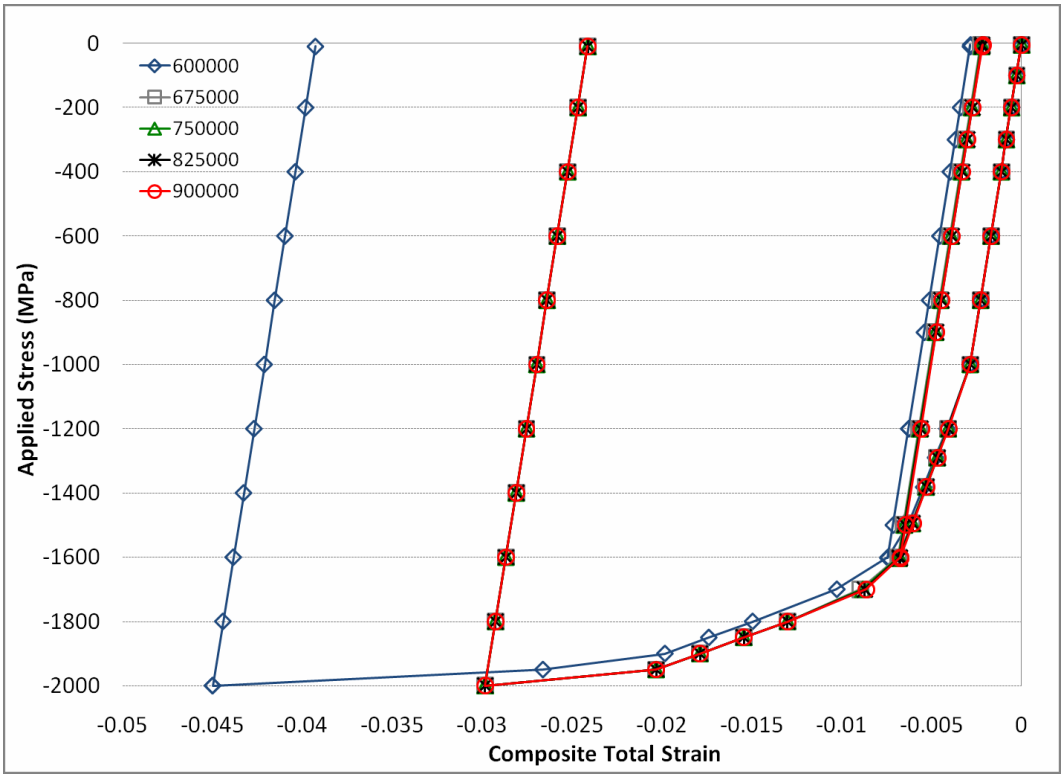


Figure 4.23 Influence of θ_0 on the composite total strain for the 80% composite as predicted by FEM

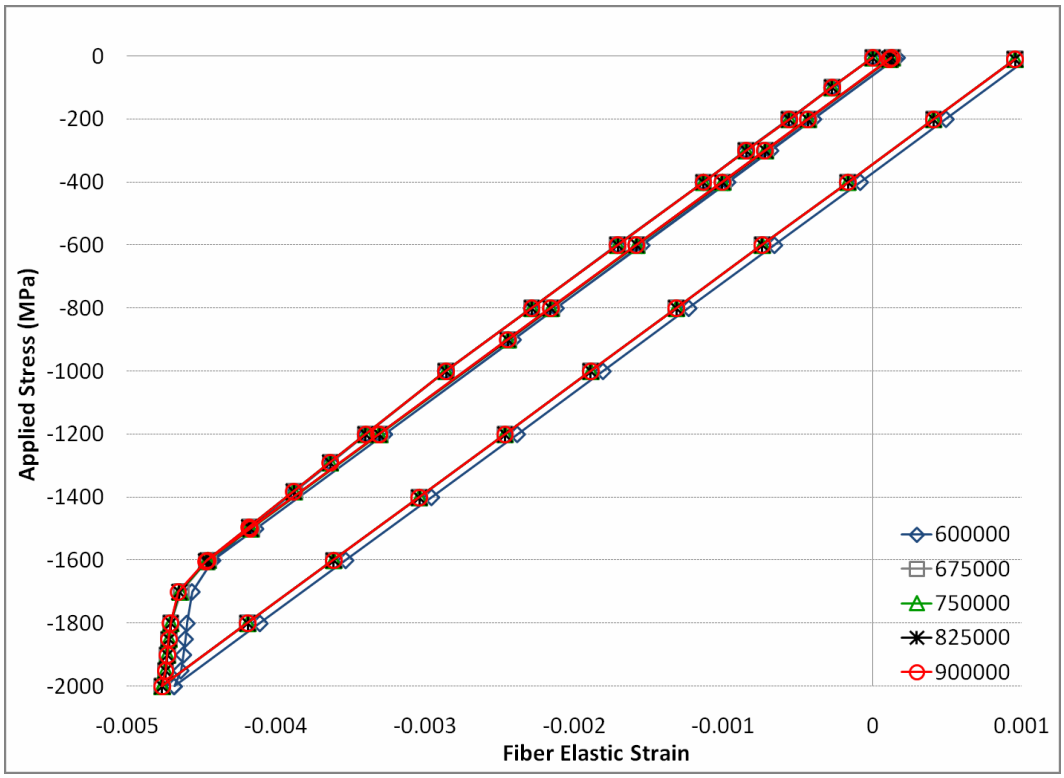


Figure 4.24 Influence of θ_0 on the fiber elastic strain for the 80% composite as predicted by FEM

4.5 Influence of θ_1

θ_1 denotes the slope of the stress strain curve for the final hardening of the fibers. (Figure 1.1) Table 4.6 shows the values of the seven parameters at each iteration to analyze the influence of θ_1 on the constitutive behavior of the composites.

Figures 4.25 through 4.32 show that θ_1 has negligible influence on the fiber elastic strain. θ_1 also has negligible influence on the composite total strain for the 20% and 40% composites. On the other hand, it has considerable influence on the composite total strain for the 60% and 80% composites during the final unloading portion of the stress-strain curves.

Table 4.6 Change in θ_1

#	σ_0^W **(MPa)**	σ_1 **(MPa)**	θ_0 **(MPa)**	θ_1 **(MPa)**	σ_0^{BMG} **(MPa)**	**n**	**T (°C)**
1	1250	700	750000	1200	2000	8	300
2	1250	700	750000	1260	2000	8	300
3	1250	700	750000	1320	2000	8	300
4	1250	700	750000	1380	2000	8	300
5	1250	700	750000	1450	2000	8	300

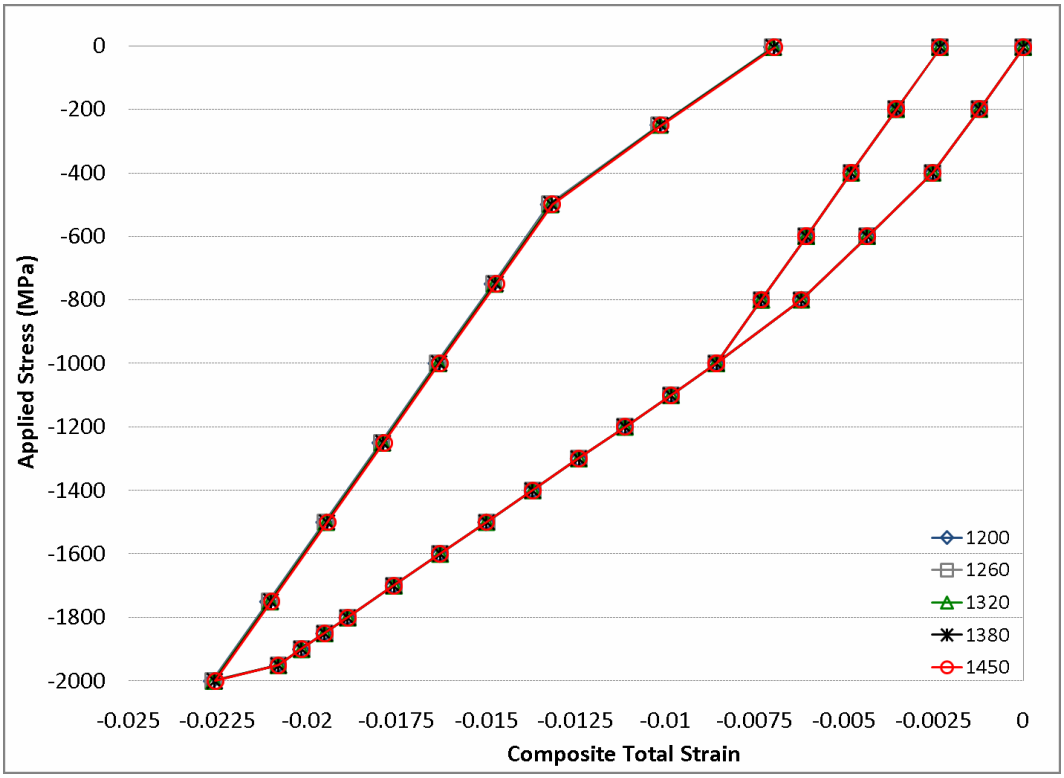


Figure 4.25 Influence of θ_1 on the composite total strain for the 20% composite as predicted by FEM

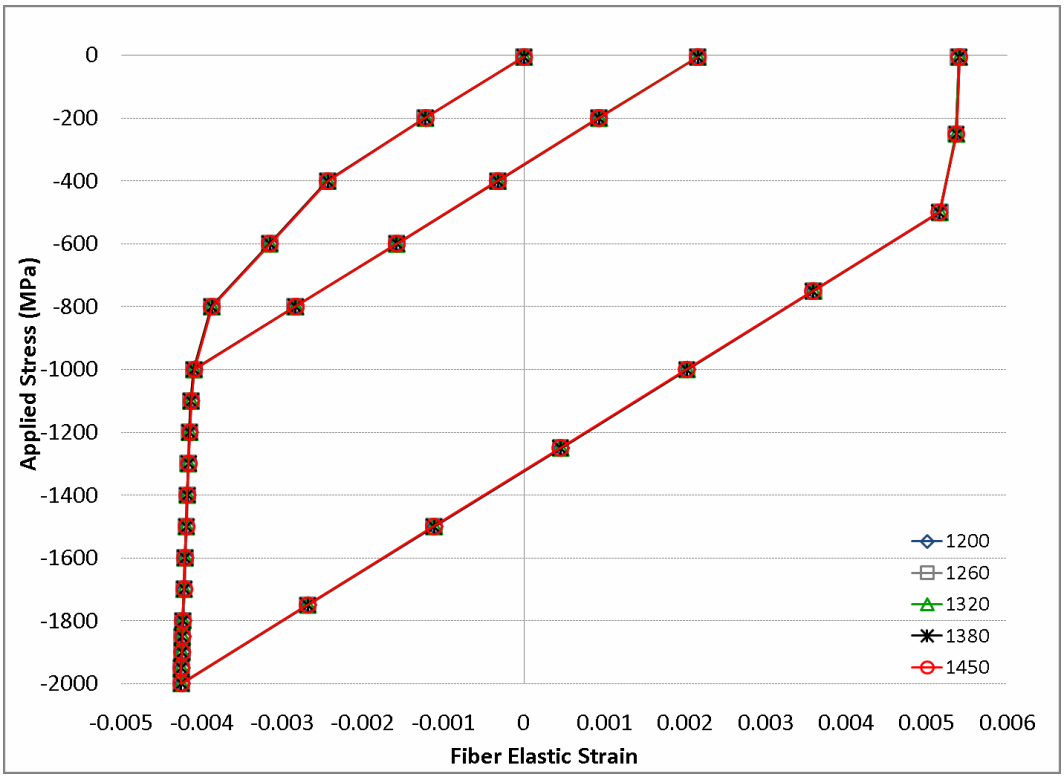


Figure 4.26 Influence of θ_1 on the elastic fiber strain for the 20% composite as predicted by FEM

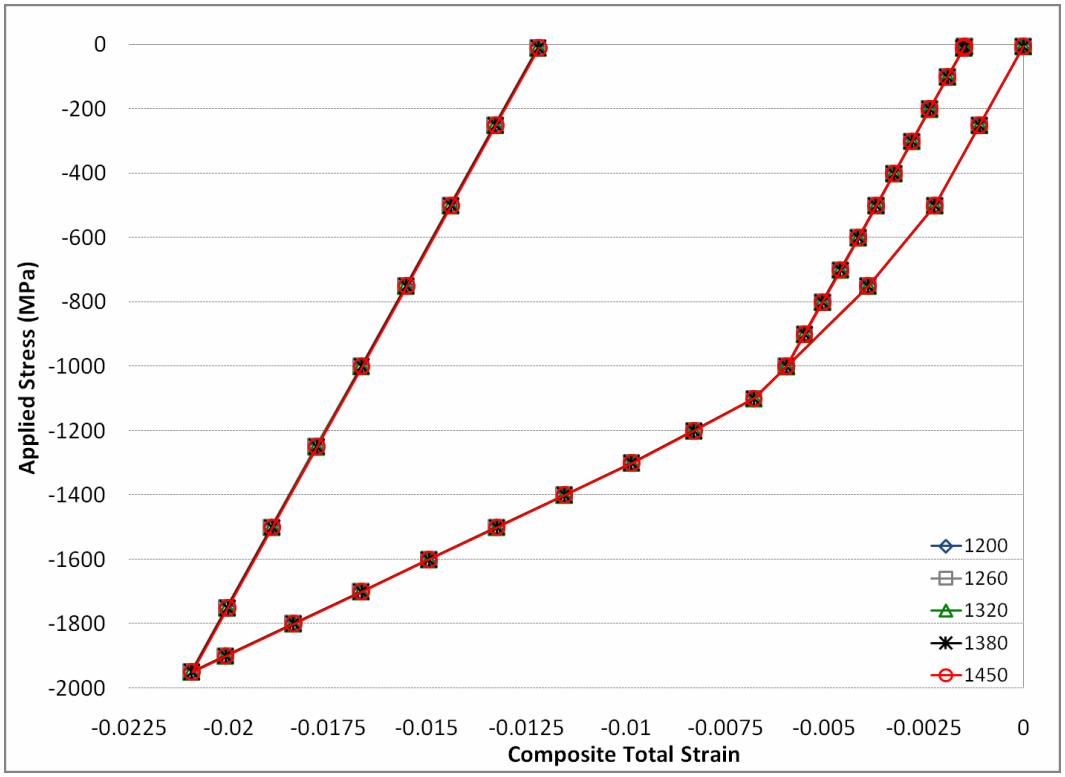


Figure 4.27 Influence of θ_1 on the composite total strain for the 40% composite as predicted by FEM

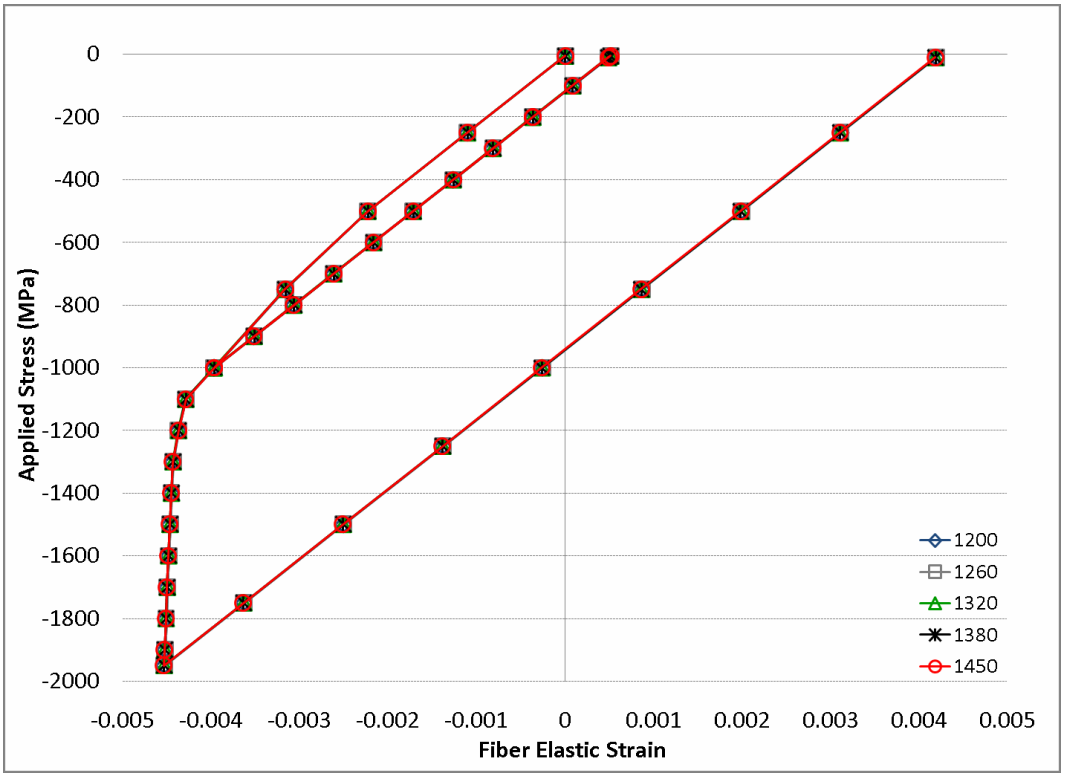


Figure 4.28 Influence of θ_1 on the elastic fiber strain for the 40% composite as predicted by FEM

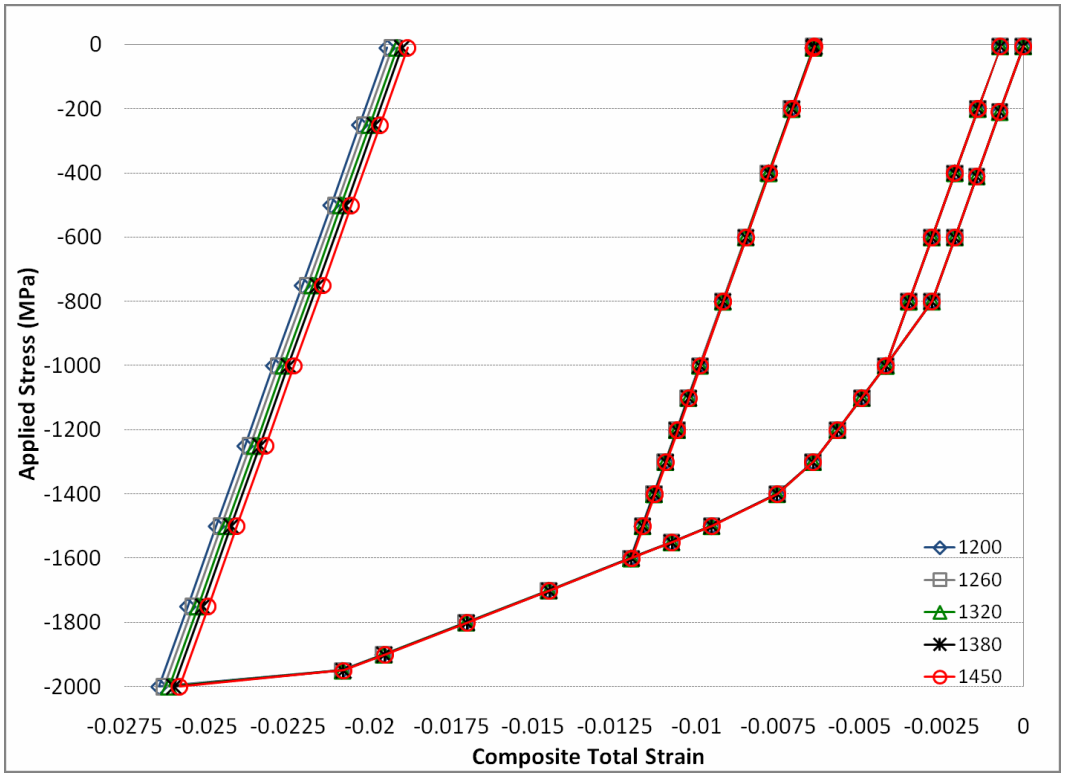


Figure 4.29 Influence of θ_1 on the composite total strain for the 60% composite as predicted by FEM

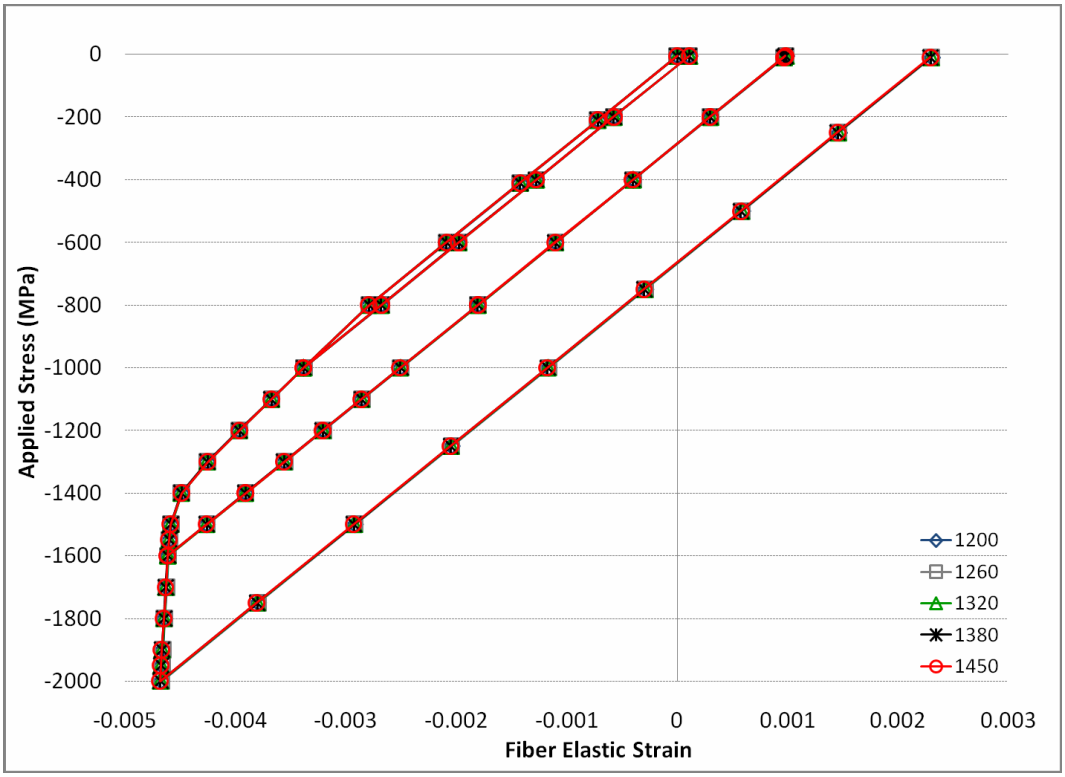


Figure 4.30 Influence of θ_1 on the elastic fiber strain for the 60% composite as predicted by FEM

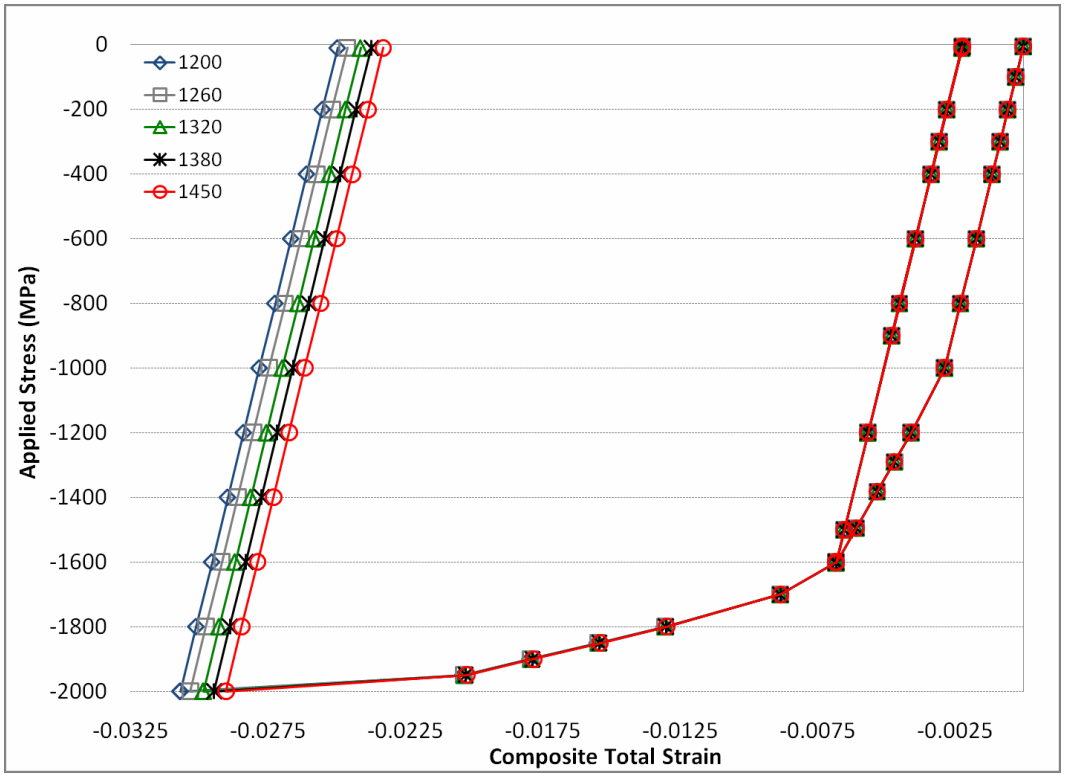


Figure 4.31 Influence of θ_1 on the composite total strain for the 80% composite as predicted by FEM

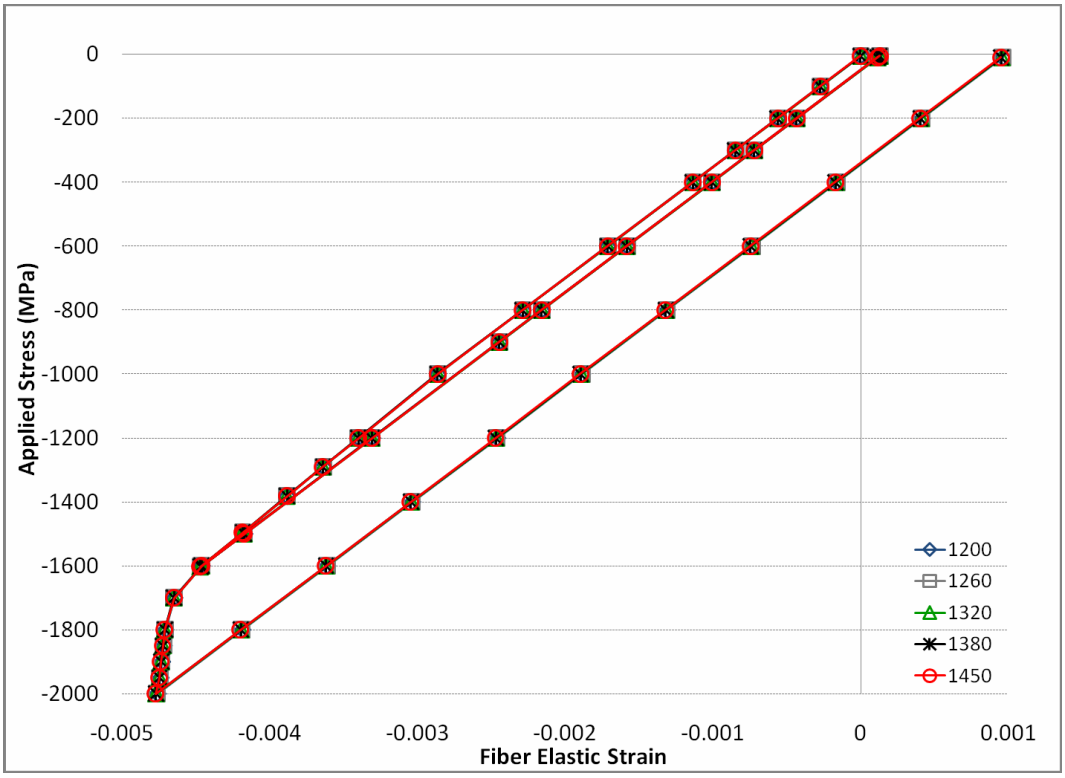


Figure 4.32 Influence of θ_1 on the elastic fiber strain for the 80% composite as predicted by FEM

4.6 Influence of the Freezing Temperature

Freezing temperature affects the amount of thermal residual stresses present in the material. Table 4.7 shows the values of the seven parameters at each iteration to analyze the influence of freezing temperature on the constitutive behavior of the composites.

Figures 4.33 through 4.40 show the influence of freezing temperature. The effect of freezing temperature on the fiber elastic strain is rather uniform and becomes progressively less pronounced with increasing fiber volume fraction. Freezing temperature also has a uniform effect on the composite total strain between 200 and 350°C. However, changing freezing temperature from 350 to 390°C has a more pronounced effect on the composite total strain than the other iterations. This can be explained by the instant change in the CTE of the BMG matrix from $9.0 \times 10^{-6} K^{-1}$ to $1.8 \times 10^{-5} K^{-1}$ around 355°C during glass transition which, in turn causes considerable thermal residual stress in the material. [5]

Table 4.7 Change in freezing temperature

#	σ_0^W **(MPa)**	σ_1 **(MPa)**	θ_0 **(MPa)**	θ_1 **(MPa)**	σ_0^{BMG} **(MPa)**	**n**	**T (°C)**
1	1250	500	750000	1320	2000	8	200
2	1250	700	750000	1320	2000	8	250
3	1250	700	750000	1320	2000	8	300
4	1250	700	750000	1320	2000	8	350
5	1250	700	750000	1320	2000	8	390

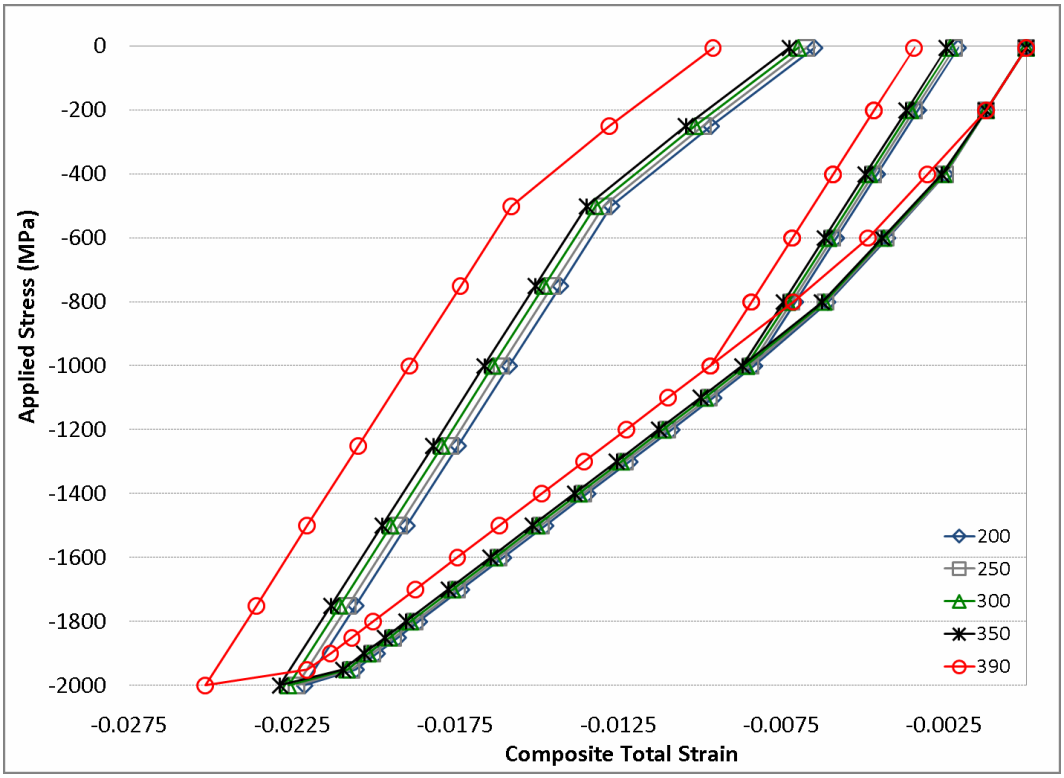


Figure 4.33 Influence of freezing temperature on the composite total strain for the 20% composite as predicted by FEM

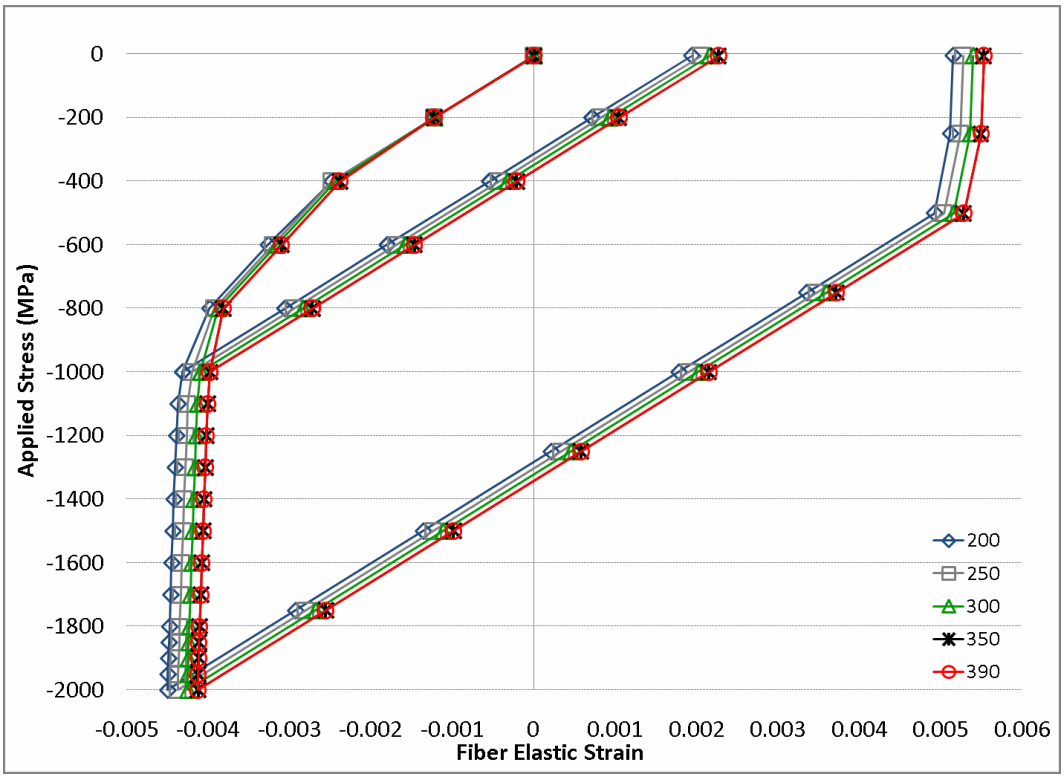


Figure 4.34 Influence of freezing temperature on the fiber elastic strain for the 20% composite as predicted by FEM

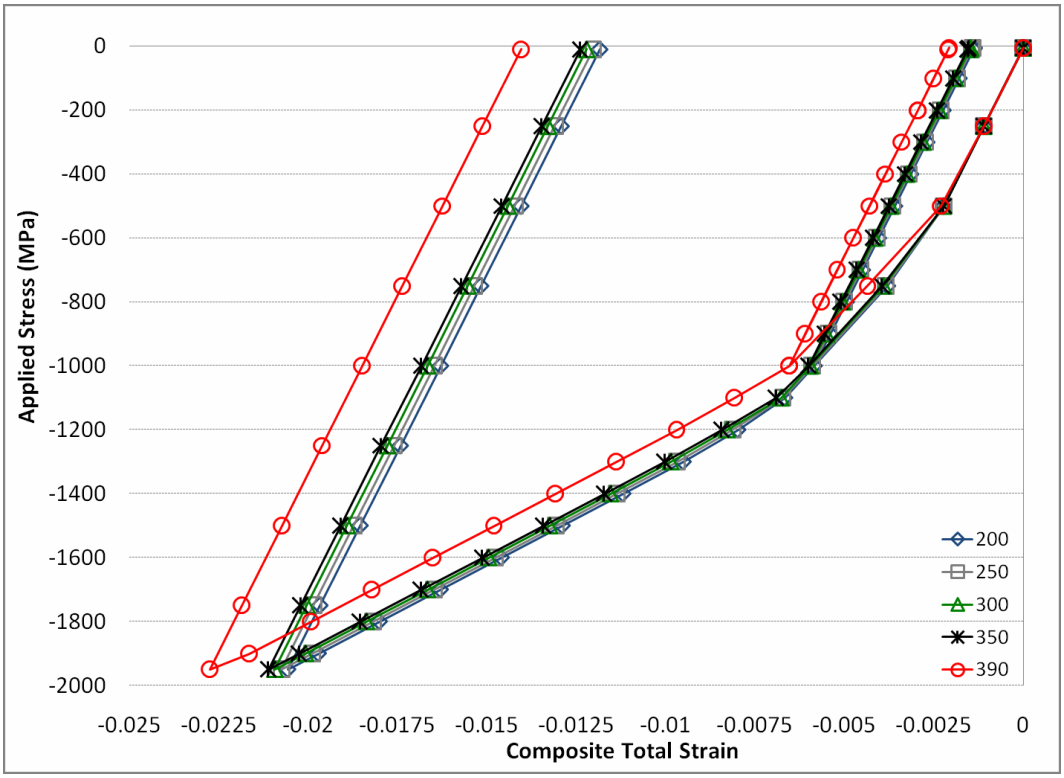


Figure 4.35 Influence of freezing temperature on the composite total strain for the 40% composite as predicted by FEM

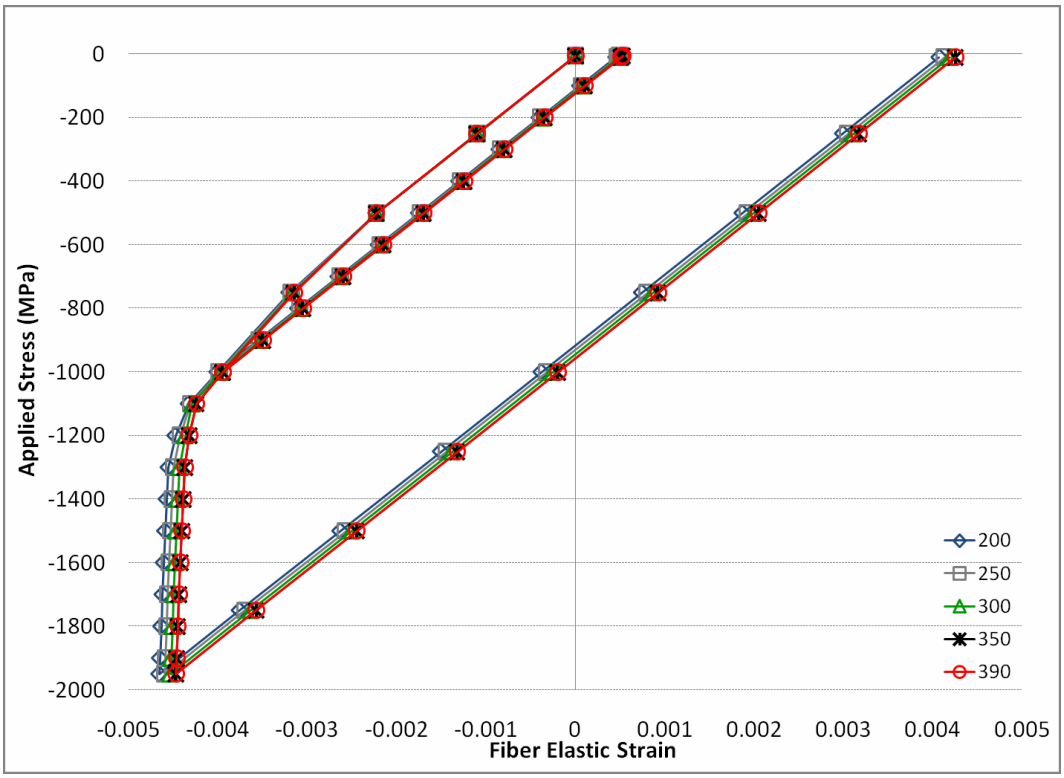


Figure 4.36 Influence of freezing temperature on the fiber elastic strain for the 40% composite as predicted by FEM

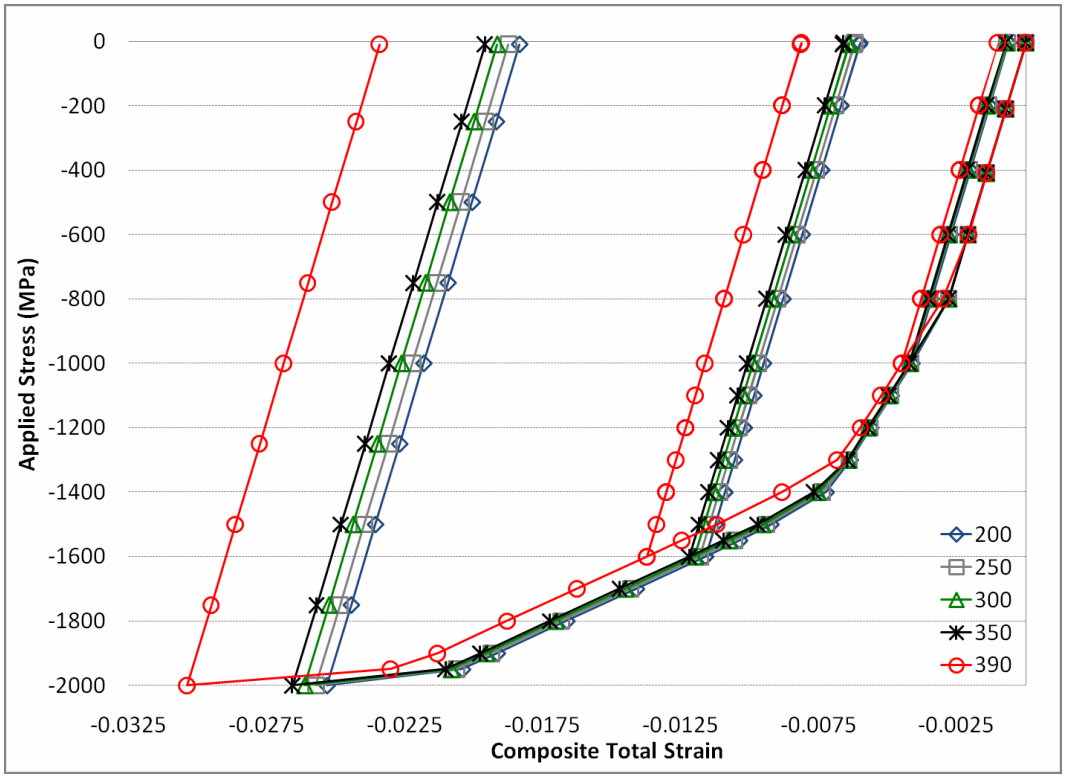


Figure 4.37 Influence of freezing temperature on the composite total strain for the 60% composite as predicted by FEM

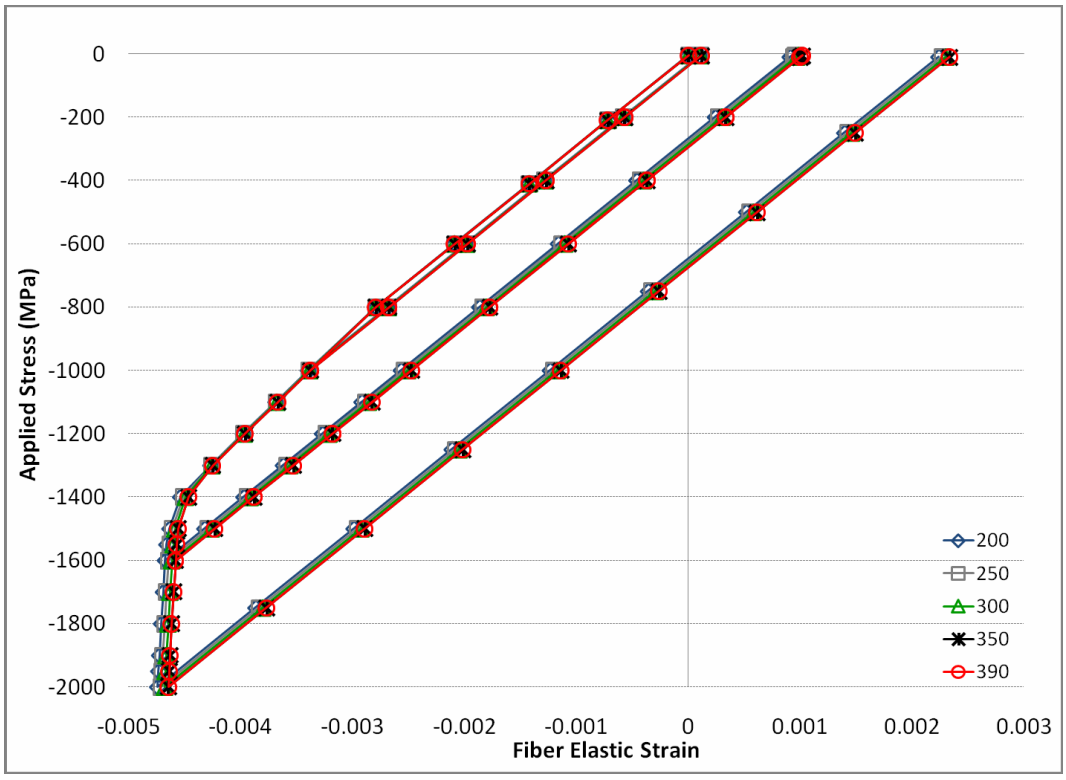


Figure 4.38 Influence of freezing temperature on the fiber elastic strain for the 60% composite as predicted by FEM

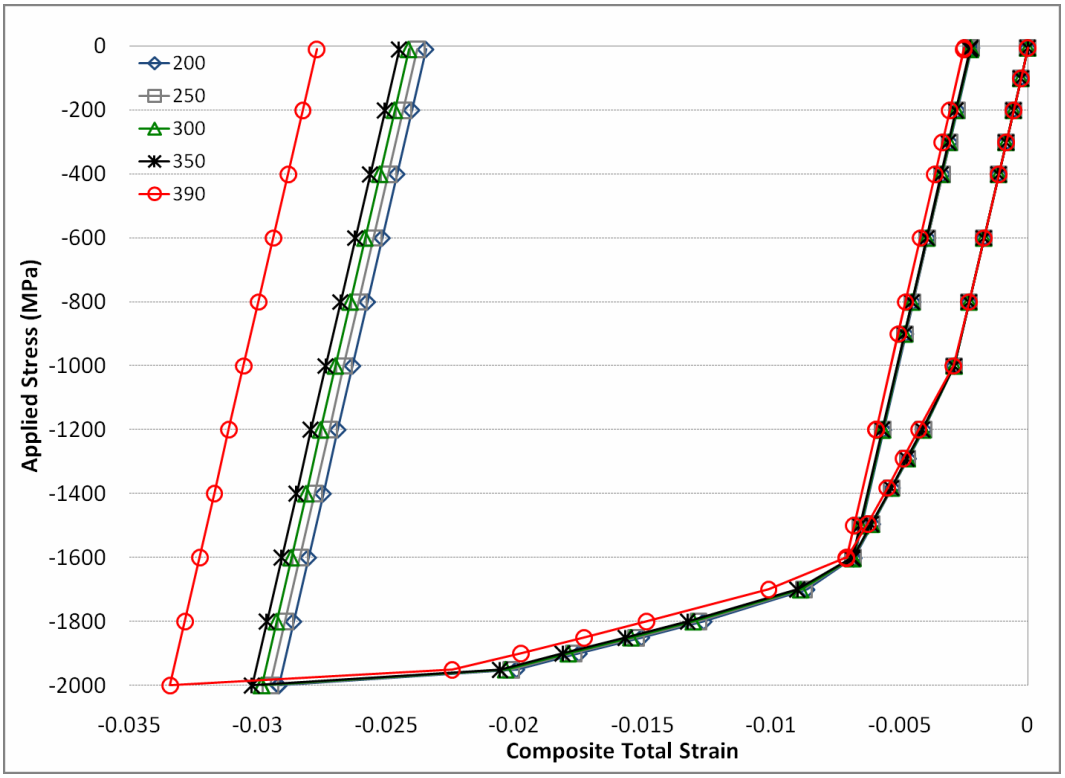


Figure 4.39 Influence of freezing temperature on the composite total strain for the 80% composite as predicted by FEM

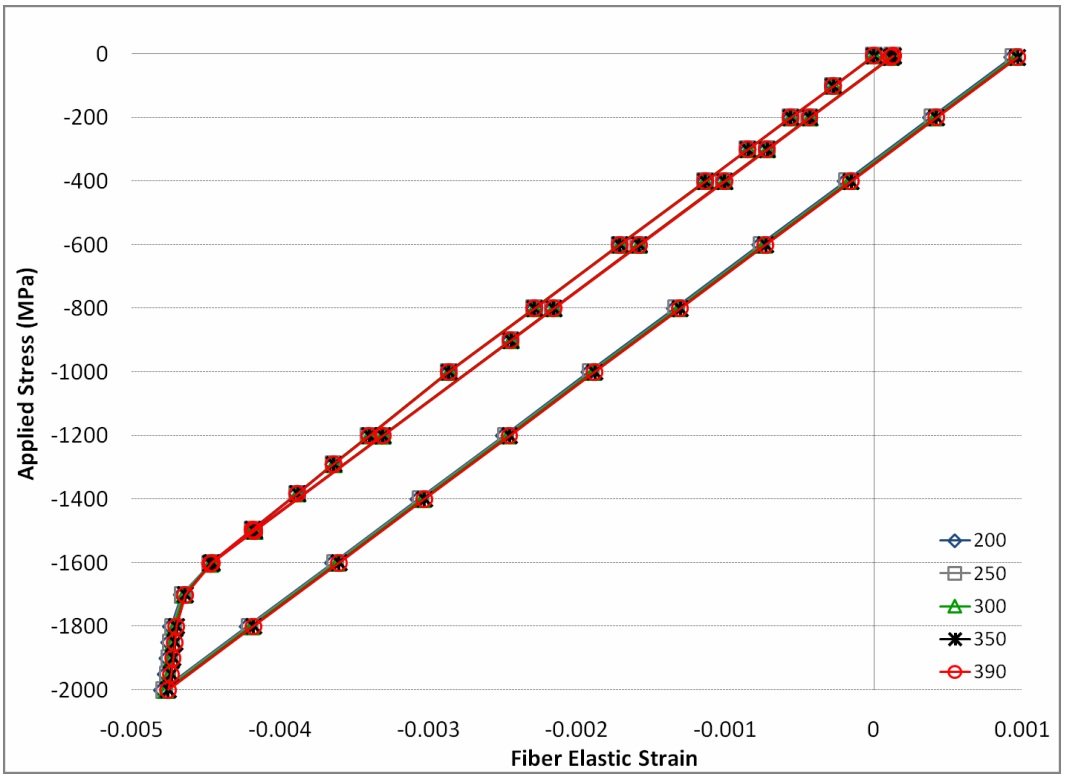


Figure 4.40 Influence of freezing temperature on the fiber elastic strain for the 80% composite as predicted by FEM

4.7 Influence of σ_0^{BMG}

σ_0^{BMG} denotes the yield strength of the BMG matrix. (Figure 1.1) Table 4.8 shows the values of the seven parameters at each iteration to analyze the influence of σ_0^{BMG} on the constitutive behavior of the composites.

Figures 4.41 through 4.48 show that σ_0^{BMG} has a major influence on the composite total strain. σ_0^{BMG} has negligible influence on the fiber elastic strain except for the 20% composite. Note that, for the 20% composite changing σ_0^{BMG} from 2050 to 2100 MPa has no effect on the composite total strain and for the 40% composite σ_0^{BMG} has a minor influence between 1950 and 2000 MPa and has no influence between 2050 and 2100 MPa. For the 60% and 80% composites σ_0^{BMG} has a stronger relationship with the composite total strain.

Table 4.8 Change in σ_0^{BMG}

#	σ_0^W **(MPa)**	σ_1 **(MPa)**	θ_0 **(MPa)**	θ_1 **(MPa)**	σ_0^{BMG} **(MPa)**	**n**	**T (°C)**
1	1250	700	750000	1320	1900	8	300
2	1250	700	750000	1320	1950	8	300
3	1250	700	750000	1320	2000	8	300
4	1250	700	750000	1320	2050	8	300
5	1250	700	750000	1320	2100	8	300

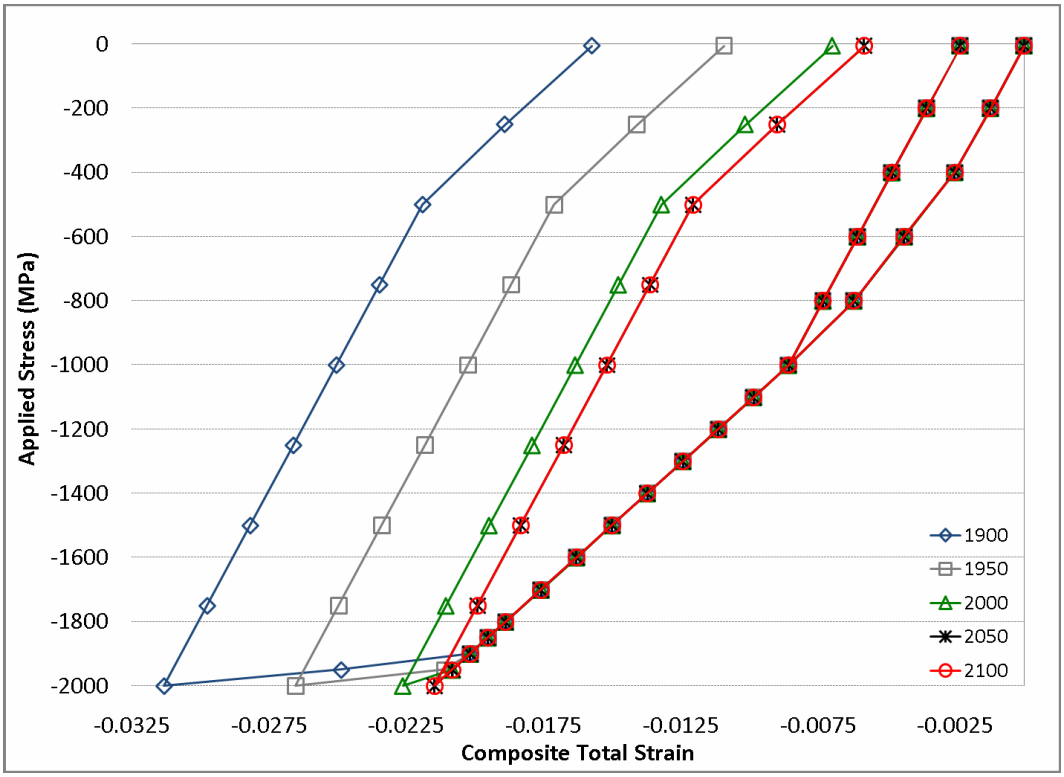


Figure 4.41 Influence of σ_0^{BMG} on the composite total strain for the 20% composite as predicted by FEM

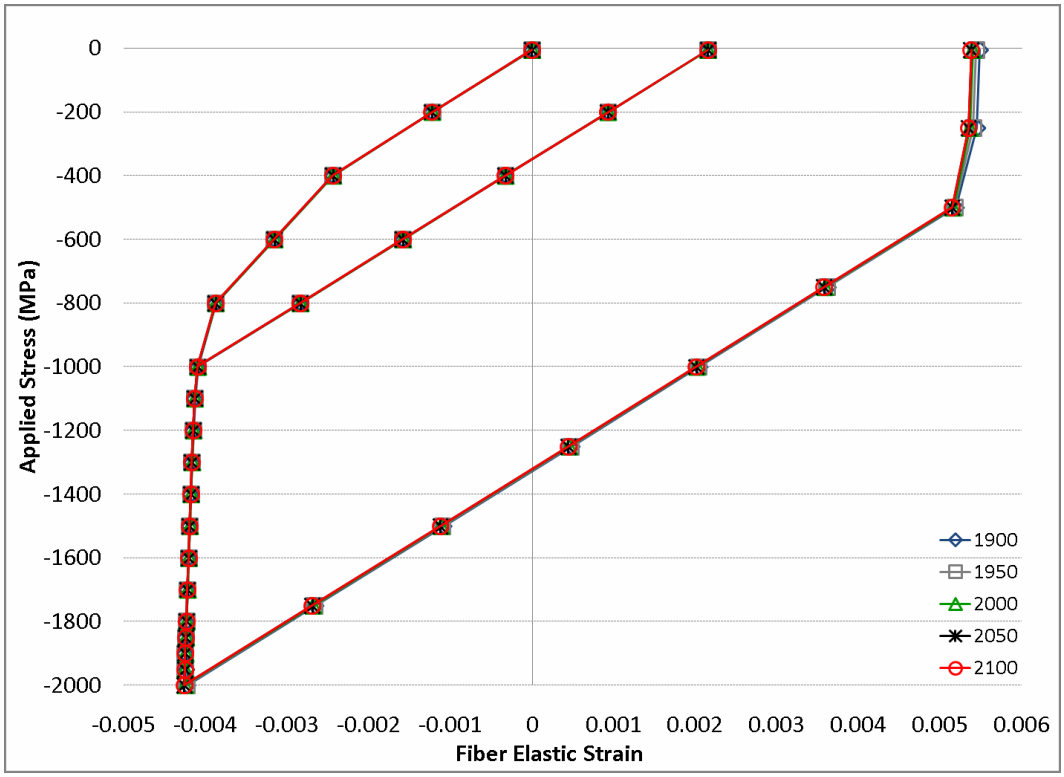


Figure 4.42 Influence of σ_0^{BMG} on the fiber elastic strain for the 20% composite as predicted by FEM

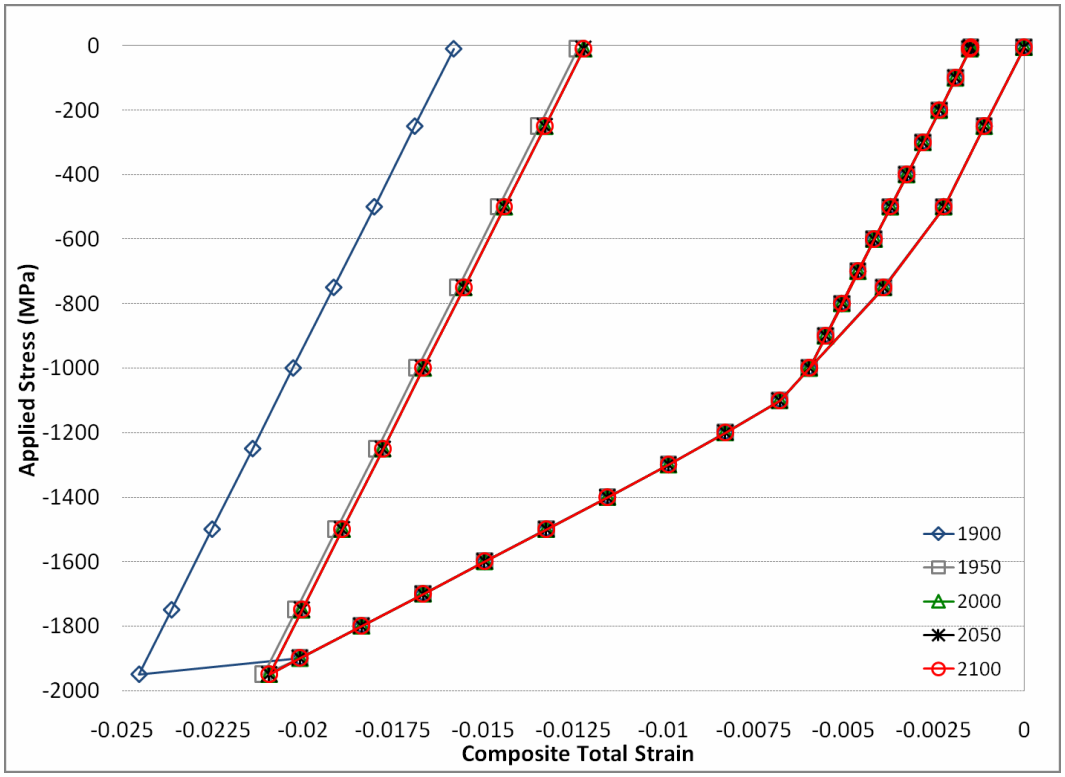


Figure 4.43 Influence of σ_0^{BMG} on the composite total strain for the 40% composite as predicted by FEM

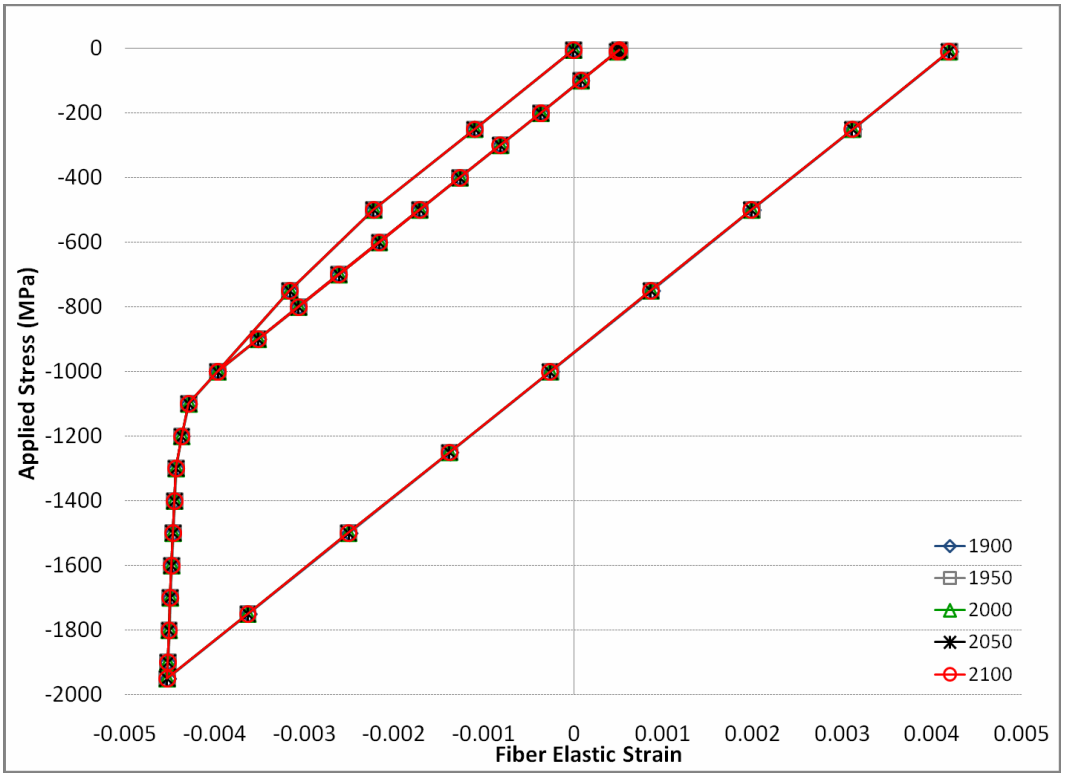


Figure 4.44 Influence of σ_0^{BMG} on the fiber elastic strain for the 40% composite as predicted by FEM

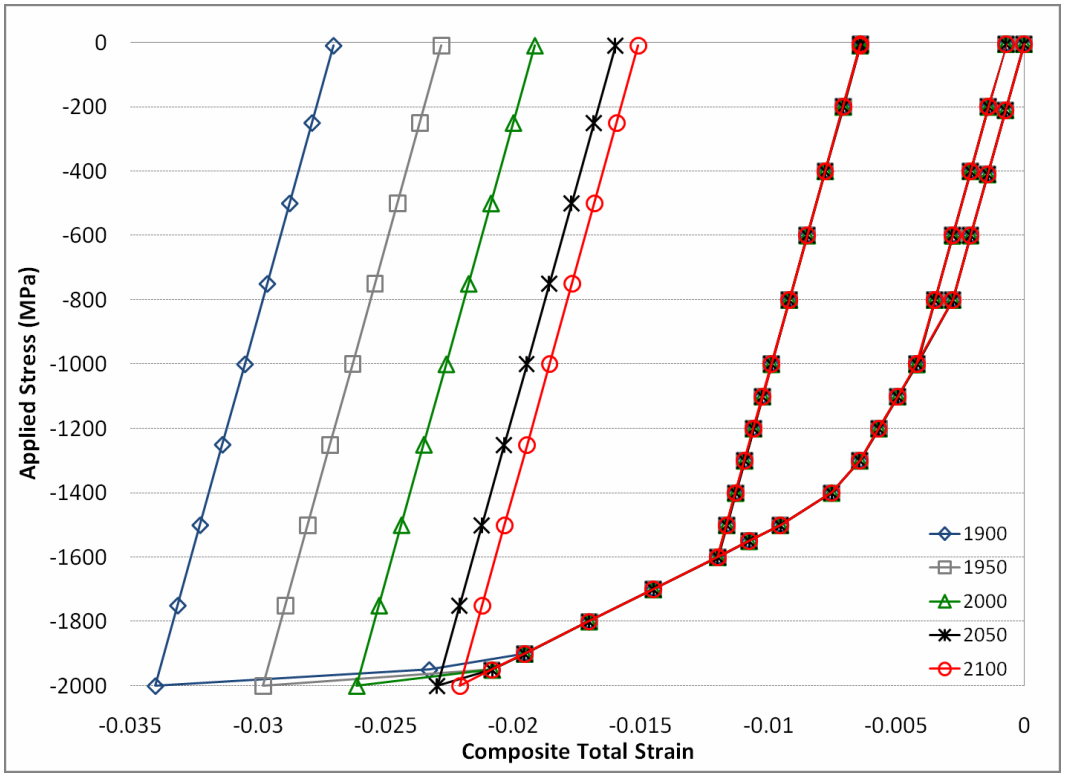


Figure 4.45 Influence of σ_0^{BMG} on the composite total strain for the 60% composite as predicted by FEM

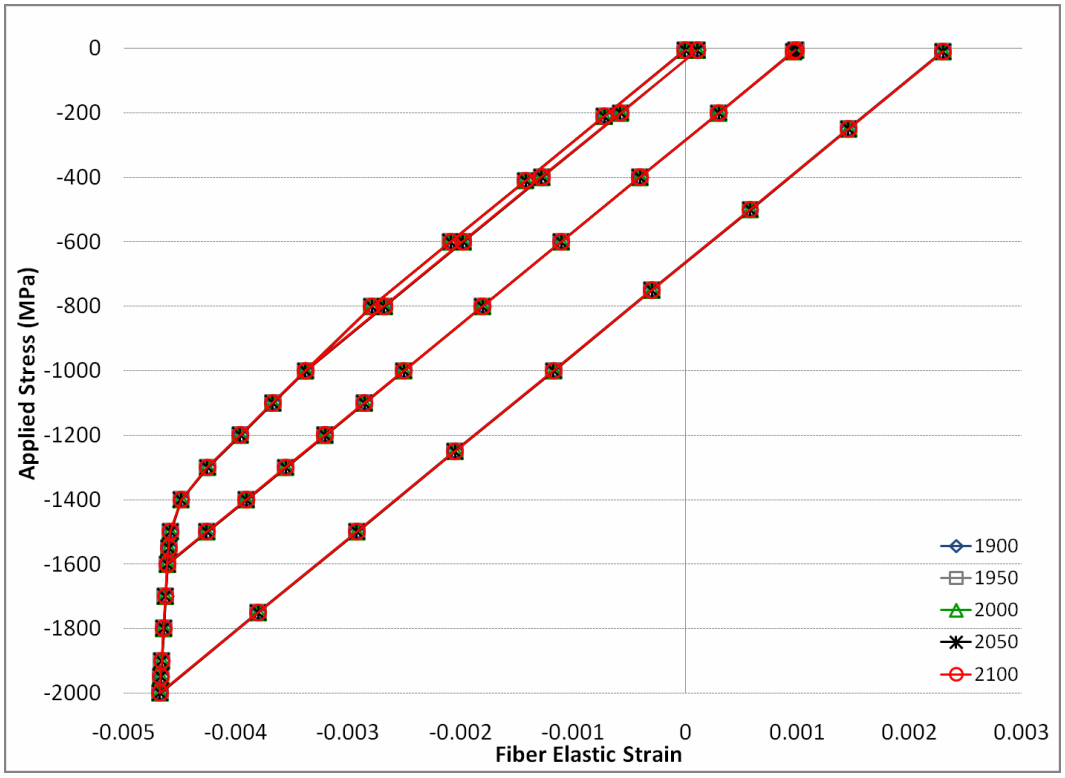


Figure 4.46 Influence of σ_0^{BMG} on the fiber elastic strain for the 60% composite as predicted by FEM

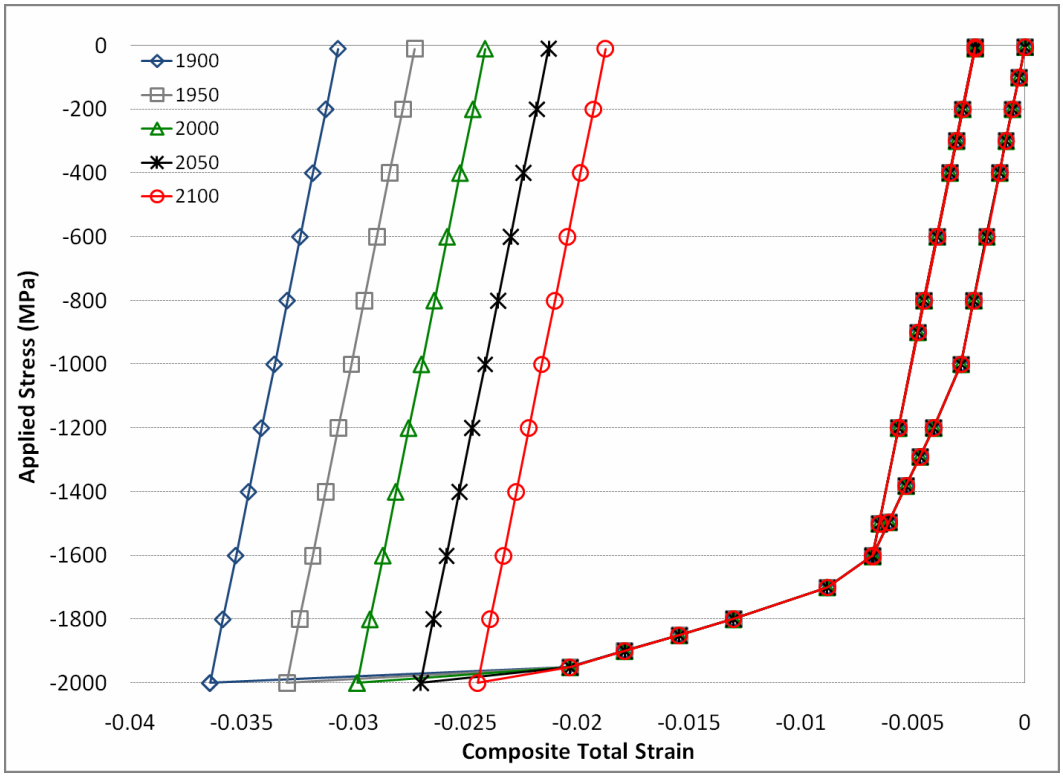


Figure 4.47 Influence of σ_0^{BMG} on the composite total strain for the 80% composite as predicted by FEM

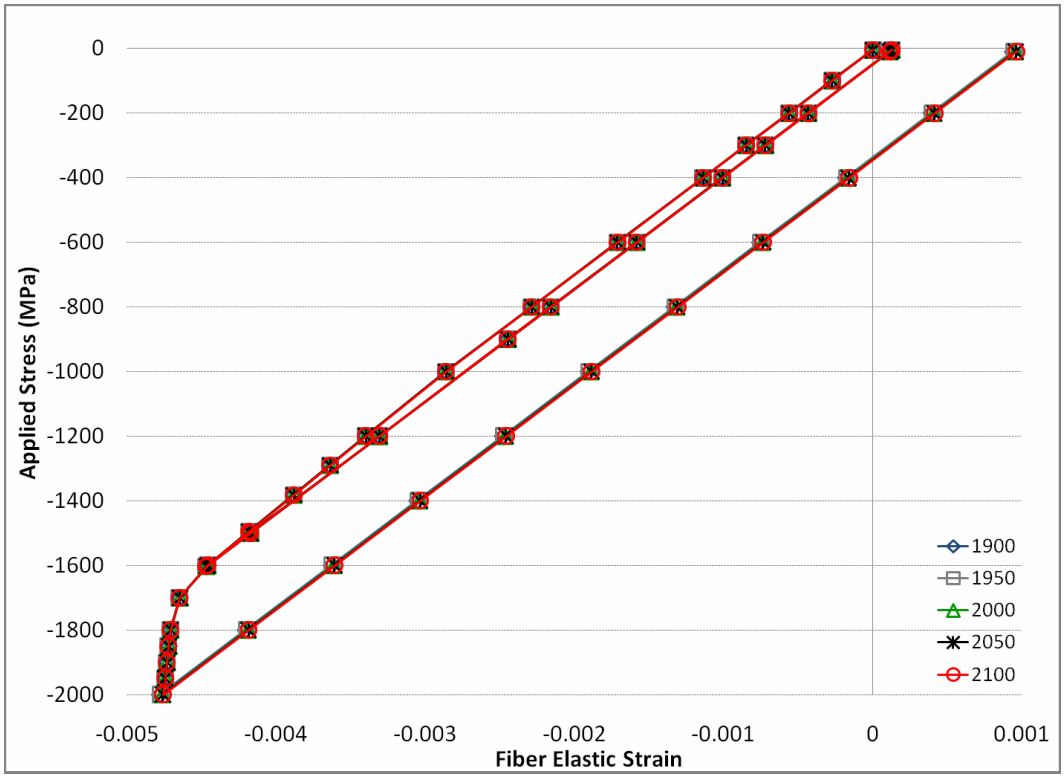


Figure 4.48 Influence of σ_0^{BMG} on the fiber elastic strain for the 60% composite as predicted by FEM

4.8 Influence of n

n denotes the power law strain hardening exponent of the BMG matrix. (Figure 1.1) Table 4.9 shows the values of the seven parameters at each iteration to analyze the influence of n on the constitutive behavior of the composites.

Figures 4.49 through 4.56 show that n has a negligible influence on the fiber elastic strain. n has a considerable influence on the composite total strain during the final unloading portion of the stress-strain curves for the 20%, 60% and 80% composites. It does not have an effect on the composite total strain for the 40% composite.

Table 4.9 Change in n

#	σ_0^W (MPa)	σ_1 (MPa)	θ_0 (MPa)	θ_1 (MPa)	σ_0^{BMG} (MPa)	n	T (°C)
1	1250	700	750000	1320	2000	2	300
2	1250	700	750000	1320	2000	5	300
3	1250	700	750000	1320	2000	8	300
4	1250	700	750000	1320	2000	11	300
5	1250	700	750000	1320	2000	15	300

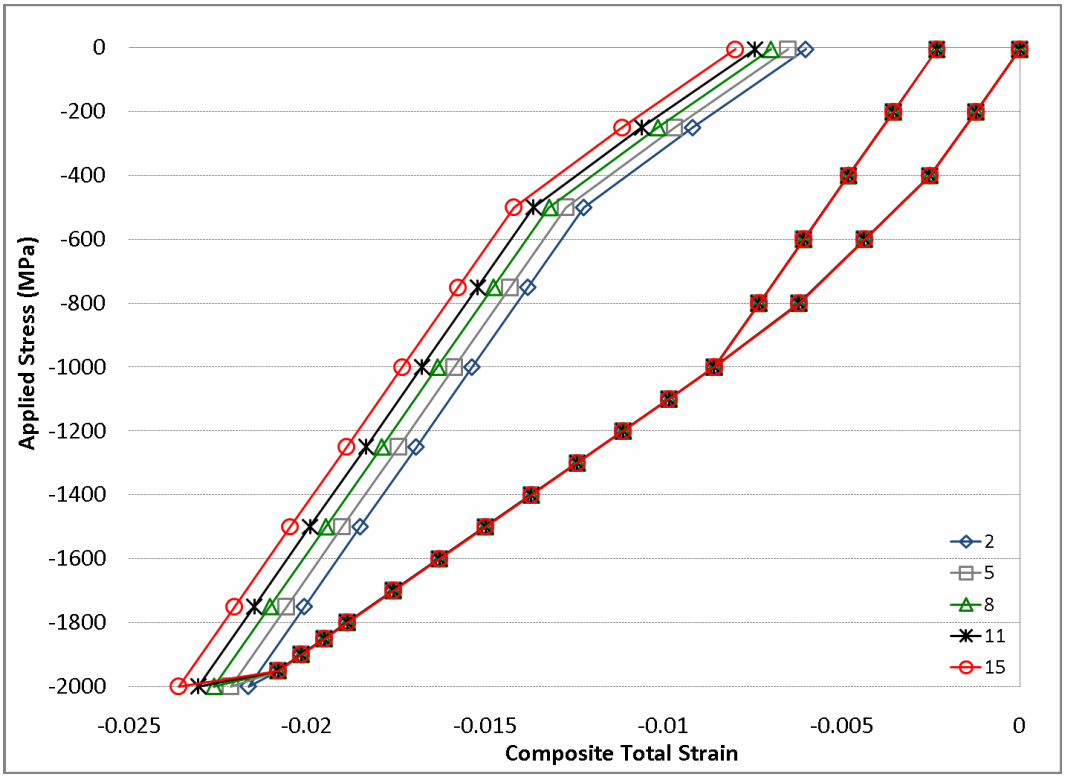


Figure 4.49 Influence of n on the composite total strain for the 20% composite as predicted by FEM

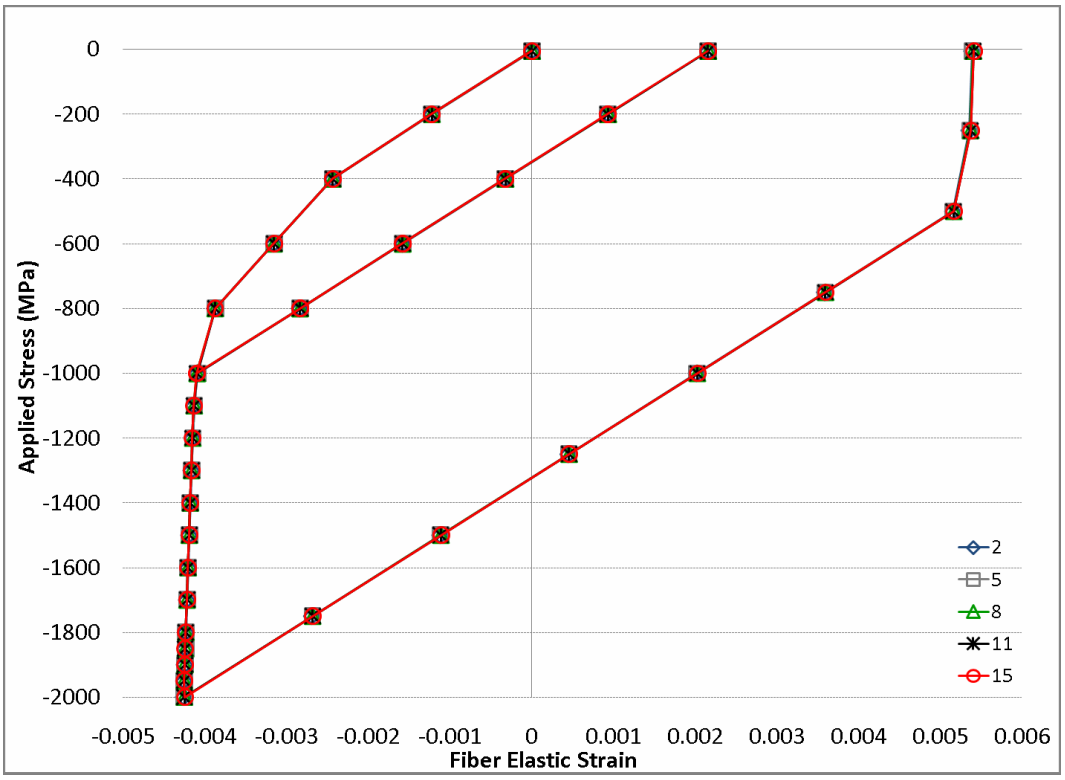


Figure 4.50 Influence of n on the fiber elastic strain for the 20% composite as predicted by FEM

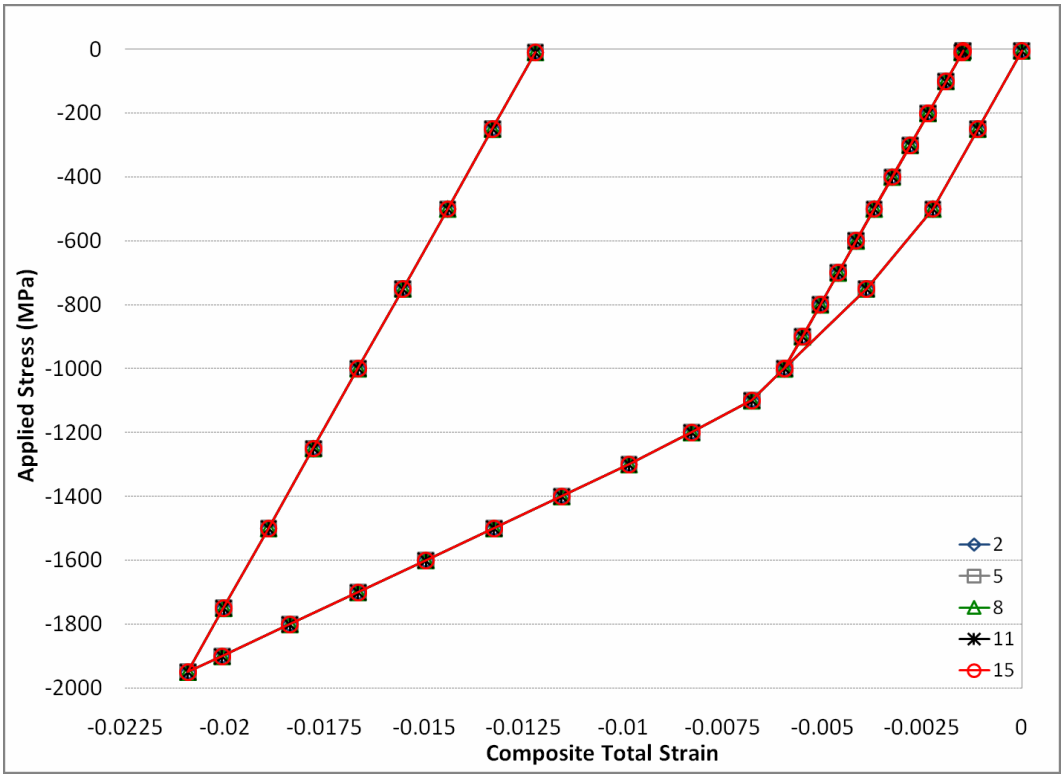


Figure 4.51 Influence of n on the composite total strain for the 40% composite as predicted by FEM

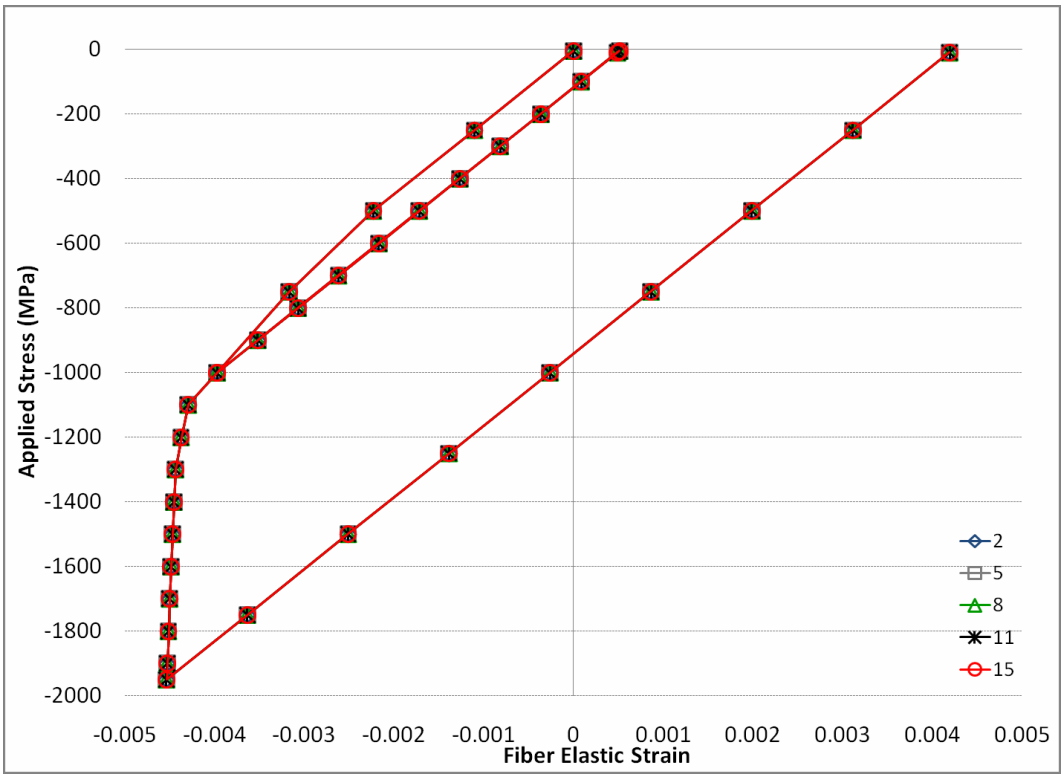


Figure 4.52 Influence of n on the fiber elastic strain for the 40% composite as predicted by FEM

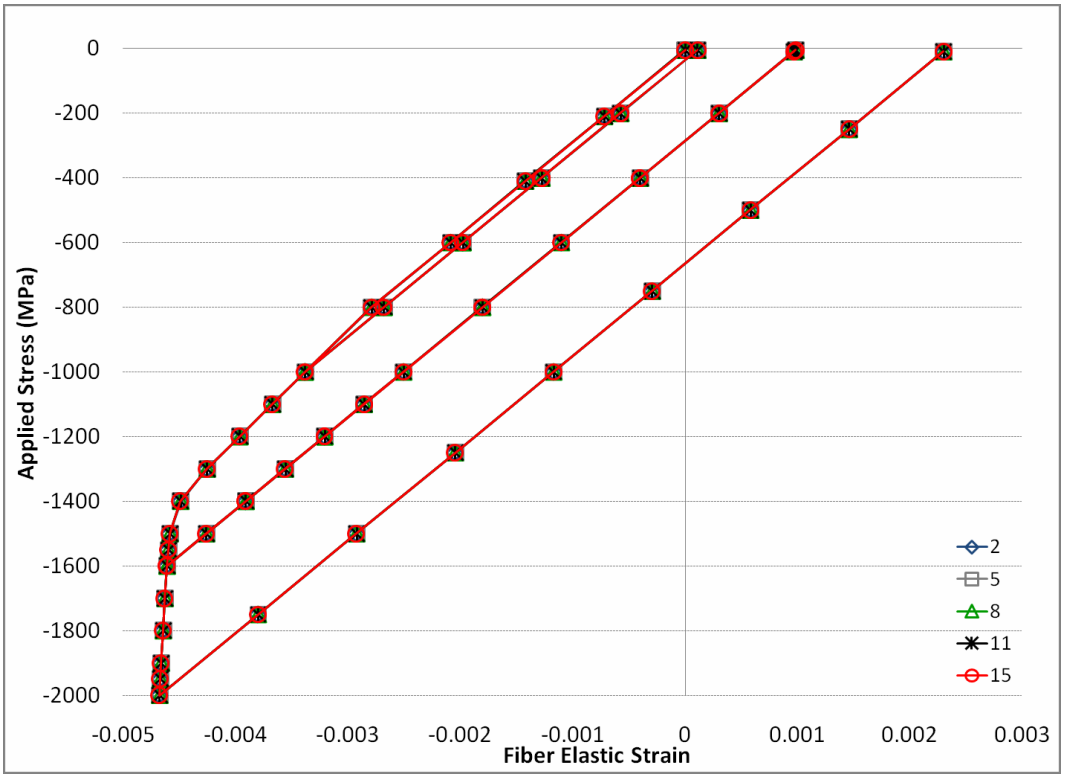


Figure 4.53 Influence of n on the composite total strain for the 60% composite as predicted by FEM

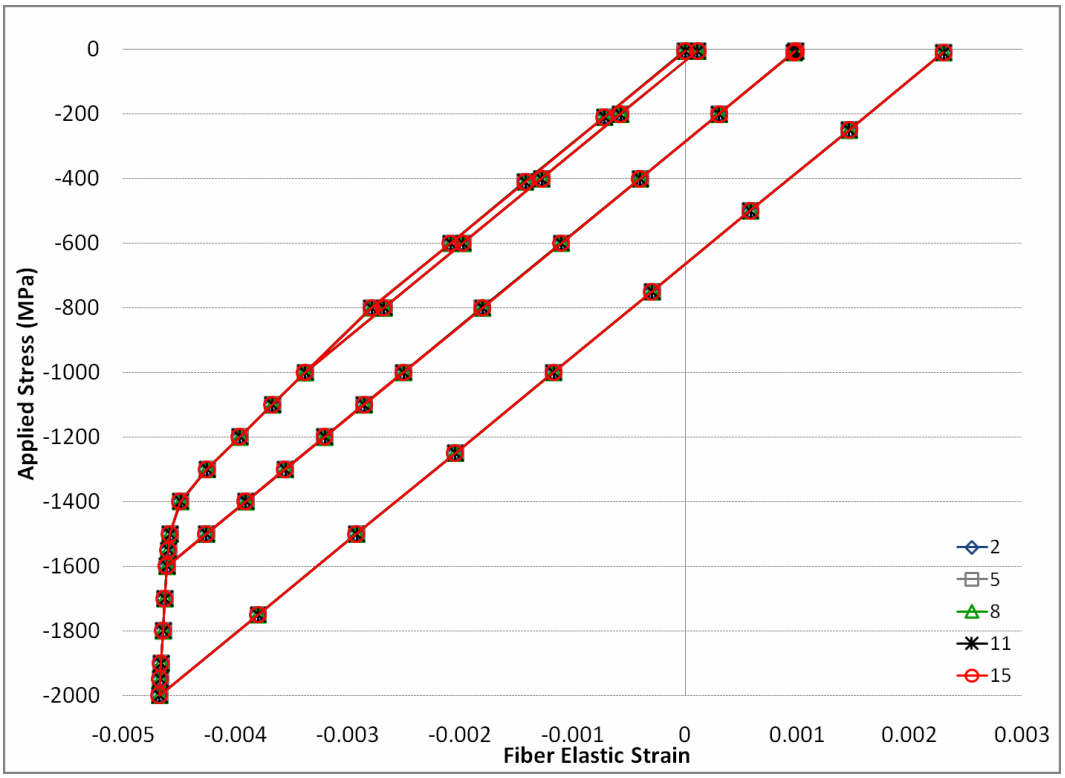


Figure 4.54 Influence of n on the fiber elastic strain for the 60% composite as predicted by FEM

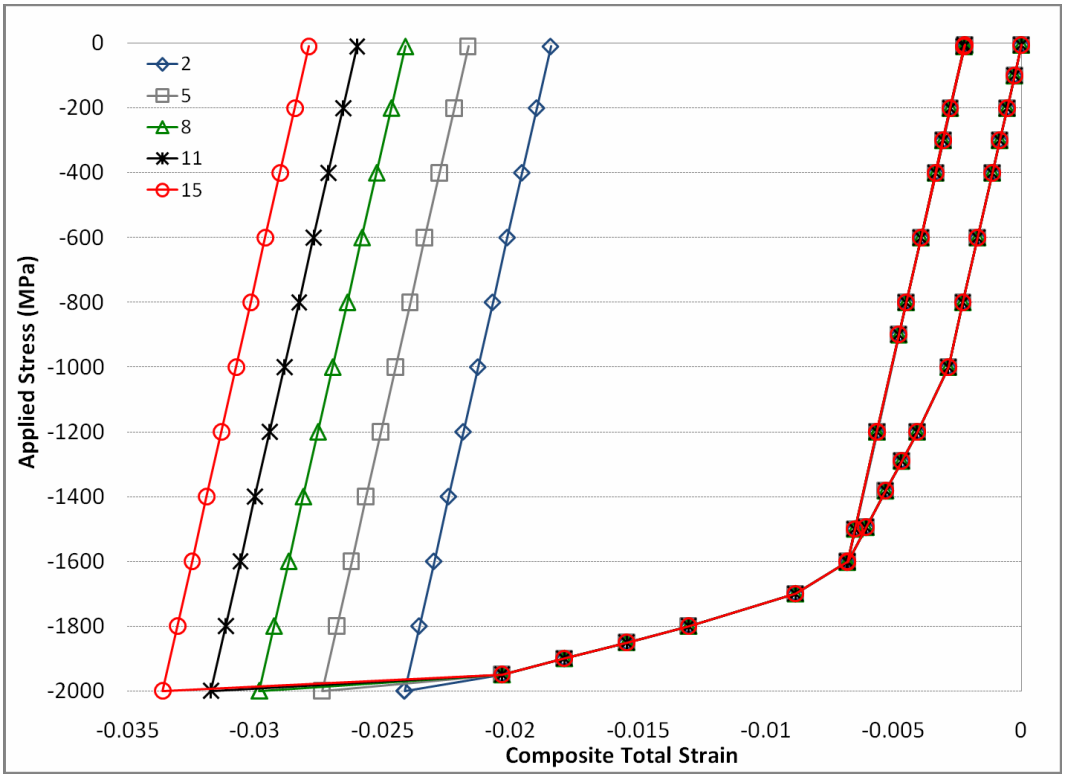


Figure 4.55 Influence of n on the composite total strain for the 80% composite as predicted by FEM

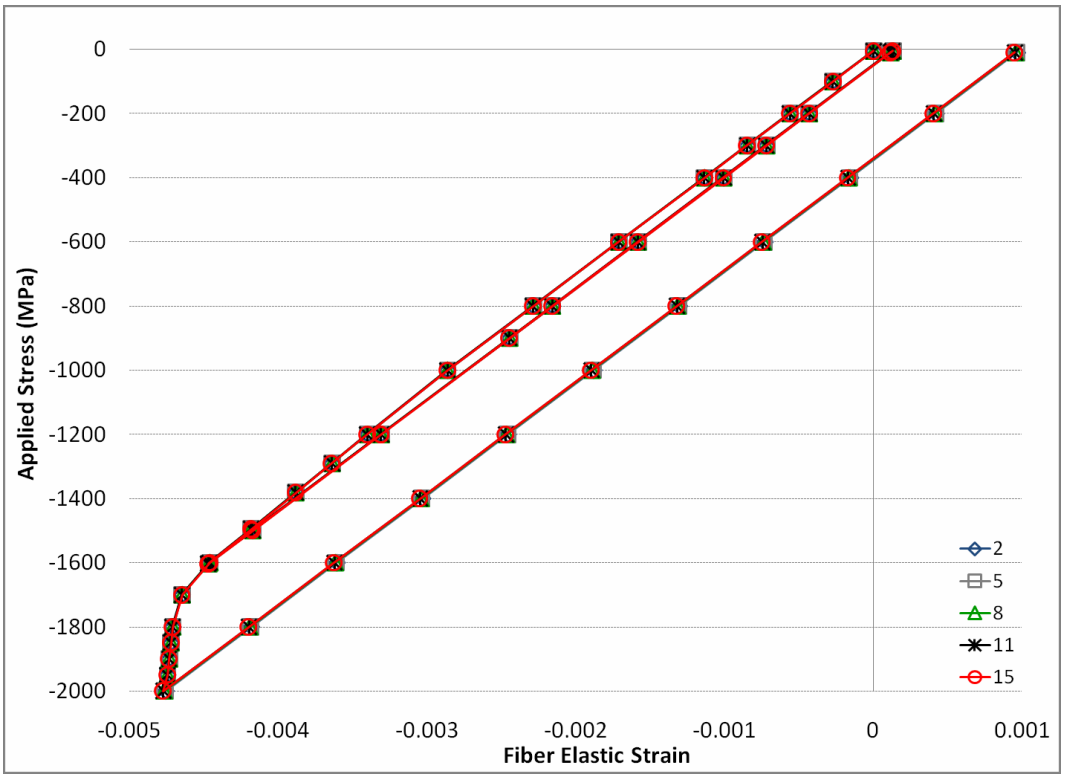


Figure 4.56 Influence of n on the fiber elastic strain for the 60% composite as predicted by FEM

4.9 Conclusions of the Sensitivity Study

The seven parameters analyzed are categorized in four groups according to their sensitivity – high, medium, low, and none.

Table 4.10 shows the results of the parametric sensitivity study for the composite total strain. σ_0^W and σ_1 are the most sensitive parameters for all composites. Although θ_0 has no sensitivity for the 20% composite it gains limited influence on the composite total strain with increasing volume fraction. In a similar manner θ_1 has no sensitivty for the 20% and 40% composites. It gains limited sensitivity for the 60% composite and a considerable sensitivity for the 80% composite. Freezing temperature has medium sensitivity for all composites. The sensitivity of σ_0^{BMG} and n strongly depend on fiber volume fraction.

Table 4.10 Parametric sensitivity for composite total strain

Composite	σ_0^W	σ_1	θ_0	θ_1	**T**	σ_0^{BMG}	**n**
20%	high	high	none	none	medium	high	medium
40%	high	high	none	none	medium	medium	none
60%	high	high	low	medium	medium	medium	none
80%	high	high	medium	medium	medium	high	high

Table 4.11 shows the results of the parametric sensitivity study for the fiber elastic strain. σ_0^W has the highest sensitivity on fiber elastic strain. It is less influential for the 60% and 80% composites than it is for the 20% and 40% composites. σ_1 is the second most sensitive parameter, although it becomes less sensitive for the 80% composite than the other composites. θ_0 has very low sensitivity for the 20% and 40% composites and it becomes marginally more sensitive for the 60% and 80% composites. θ_1 has no sensitivity on fiber elastic strain. The effect of freezing temperature becomes less pronounced with increasing fiber volume fraction. σ_0^{BMG} has low sensitivity for the 20% and 80% composites and has no sensitivity for the 40% and 60% composites. The power law exponent has no influence on the fiber elastic strain.

Table 4.11 Parametric sensitivity for fiber elastic strain

Composite	σ_0^W	σ_1	θ_0	θ_1	**T**	σ_0^{BMG}	**n**
20%	high	high	none	none	medium	low	none
40%	high	high	none	none	medium	none	none
60%	high	high	low	none	medium	none	none
80%	high	medium	medium	none	low	low	none

The parametric sensitivity study shows that the sensitivity of the parameters strongly depends on fiber volume fraction. The sensitivity of the parameters are also different for the fiber elastic strain and the composite total strain.

The results of the sensitivity study has very important consequences in developing forward and inverse neural network models. For the forward neural network models the insensitive parameters should not be included in neural network training because this would deteriorate the neural network predictions. On the other hand development of an inverse neural network model for an insensitive parameter is unnecessary because a change in the parameter does not change the outcomes. Consequently, insensitive parameters will be discarded from forward neural network models and inverse neural network models will not be constructed for insensitive parameters in this study.

CHAPTER 5. FORWARD NEURAL NETWORK MODELS

5.1 Introduction

In the present study, the goal of a forward neural network is to predict elastic lattice strains in the W fibers as well as the total (macroscopic) composite strains at the same loading points as those employed during the neutron diffraction experiment of the composite. The inputs to the neural network are Voce parameters (σ_0^W, σ_1, θ_0 and θ_1), power law parameters (σ_0^{BMG} and power law exponent) and freezing temperature. Figure 5.1 shows a schematic representation of the forward neural network model.

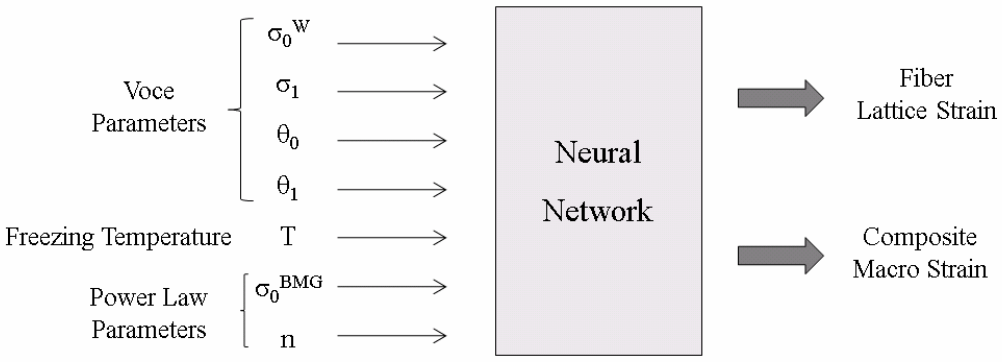


Figure 5.1 Schematic representation of the forward neural network model

5.2 Generation of Training and Testing Sets

The training and testing sets are obtained from finite element modeling using the commercial software $ABAQUS^{TM}$. The details of the finite element model (FEM) are given in chapter 1. Python programming language is used to control $ABAQUS^{TM}$ to generate loading-unloading curves by randomly changing the seven parameters governing the mechanical behavior of the materials within the ranges given in Table 5.2. Figure 5.2 shows a sample loading-unloading curve obtained from one iteration of the Python script for the 20% composite. For this particular example, the loading-unloading curve has 38 data points for the fibers and 38 data points for the composite. These loading points are the same ones also employed during the neutron diffraction experiment of this composite. The number of data points on the loading-unloading curves is different for each composite and depends on the specific loading-unloading cycle employed.

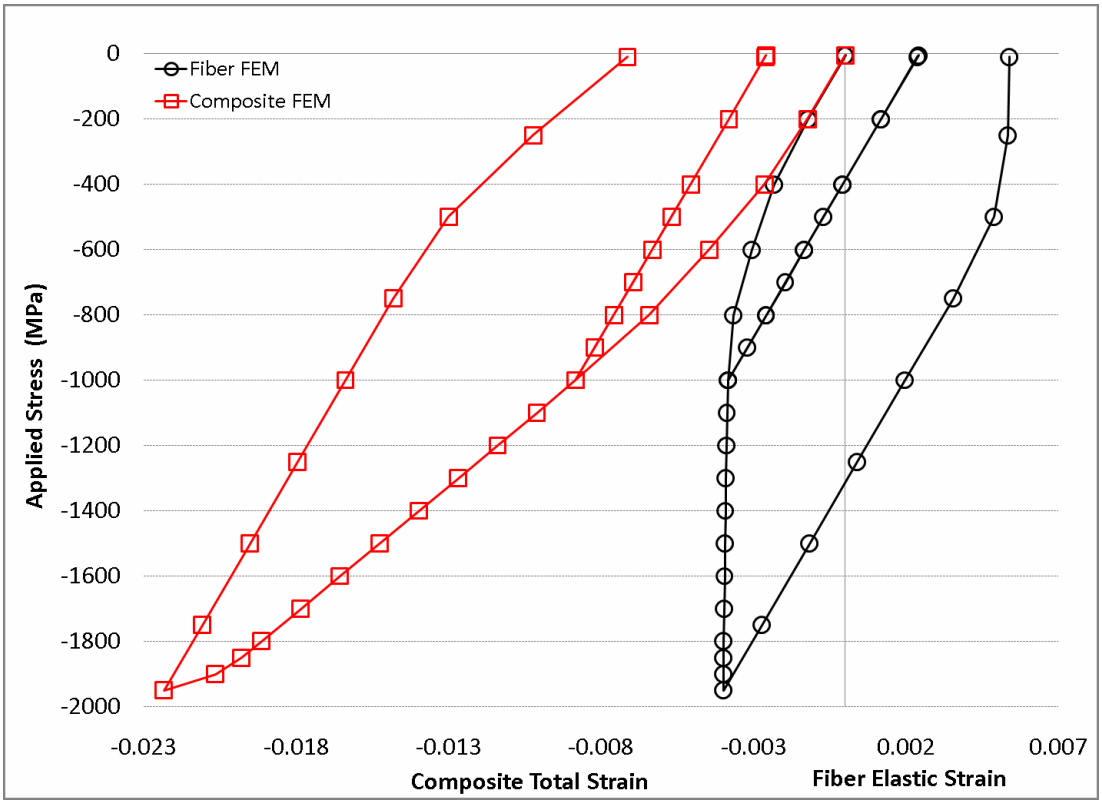


Figure 5.2 Sample FEM loading-unloading curve obtained for the 20% composite

Note that, the first data point on both the fiber elastic strain and the composite total strain loading-unloading curves is (0,0). This is counter-intuitive when the presence of thermal residual strains in the fibers is considered. However, during the neutron diffraction experiment a -5 MPa load is initially applied to the sample to hold it in place prior to the application of the loading-unloading cycle and all strains are measured relative to that initial stress state. Therefore, while the FEM does calculate thermal residual strains in both phases (as a function of the freezing temperature), its results are modified to plot the strains in the fibers and the composite relative to that particular initial stress state to be able to simulate the experiment accurately.

Table 5.1 shows the number of loading-unloading curves generated by FEM for each composite. The loading-unloading curves are partitioned into training and testing sets. Although, the number of examples in the training sets vary for each composite, the number of testing examples is fixed at 500. The variation in training set size is caused by the different number of non-converging FEM runs during the generation of the loading-unloading curves.

Table 5.1 Number of training and testing examples generated by FEM for each composite

Composite	**Total**	**Training Set**	**Testing Set**
20%	3528	3028	500
40%	4954	4454	500
60%	4341	3841	500
80%	4874	4374	500

Table 5.2 Ranges of Voce and power law parameters, and the freezing temperature. (Chapter 1)

Variable	**Unit**	**Min**	**Max**
σ_0^W	MPa	1000	1500
σ_1	MPa	500	800
θ_0	MPa	600000	900000
θ_1	MPa	1200	1450
σ_0^{BMG}	MPa	1900	2100
n	–	2	15
Temperature	°C	200	390

Training and testing sets are normalized before they are presented to the neural network. The inputs to the neural network comprise of seven independent parameters each varying in a different range. (Table 5.2) Each input dataset is normalized to between -2.5 and +2.5 from their individual ranges. The normalized values of the parameters are calculated by using equation 5.1

$$N = \left[\left(\frac{P - P_{min}}{P_{max} - P_{min}}\right) \times (n_{max} - n_{min}) + n_{min}\right] \tag{5.1}$$

Here, N refers to the normalized value of the parameter. P refers to the value of the parameter before normalization. P_{min} refers to the minimum value of the parameter within its range. P_{max} refers to the maximum value of the parameter within its range. n_{min} refers to the minimum value of the normalization range and n_{max} refers to the maximum value of the normalization range. Refer to table 5.2 for the ranges of the input parameters. For example, for $\sigma_0 = 1200$ MPa, equation 5.1 becomes:

$$N = \left[\left(\frac{1200 - 1000}{1500 - 1000}\right) \times (2.5 - (-2.5)) + (-2.5)\right] = -0.5$$

Normalizing the inputs to the same range is a common practice to improve neural network performance. The applied range is adopted through research conducted by H. Ceylan. [62]

On the other hand, the targets consist of two partially dependent datasets both of which comprise of dependent datasets of 35 data points each. In other words, the strain points on the loading-unloading curves of the composite are related to each other, as are the strain points on the fiber loading-unloading curve. Moreover, the strains on the fibers affect the strains on the composite. In normalizing the targets, the minimum and maximum values for the fibers and the composite are determined by considering the data points on their respective curves as a group. Then the strains on the fibers and the composite are normalized to between 0.1 and 0.9 from two separate ranges using equation 5.1. For example, the maximum and minimum fiber elastic strains for the 20% composite are calculated as 0.007473 and -0.005239 respectively and

the fiber elastic strain at the last point of the loading-unloading cycle is predicted as 0.005315 by FEM. For this particular point, equation 5.1 becomes:

$$N = \left[\left(\frac{0.005315 - (-0.005239)}{0.007473 - (-0.005239)} \right) \times (0.9 - 0.1) + 0.1 \right] = 0.764187$$

As the target values will be compared to neural network predictions their ranges should be compatible with each other and the range of the sigmoid activation function (Equation 2.31) used in the output nodes. This range is chosen instead of the total range of the sigmoid function ([0, 1]) to avoid the very sharp curvature of the sigmoid function near its extrema.

5.3 Neural Network Architectures

The neural network architecture refers to the number of nodes in the input layer, the number of hidden layers and the number of neurons in each hidden layer, and number of neurons in the output layer.

The number of nodes in the input layer is determined by the sensitivity of the seven parameters governing the constitutive behavior of the composites. As including insensitive parameters in neural network inputs would cause confusion of the neural network and deteriorate the neural network predictions, insensitive parameters are discarded from the neural network models. According to the results of the sensitivity study conducted (Chapter 4) θ_0 and θ_1 were found to be insensitive for the 20% and 40% composites. Therefore, the neural network models built for the 20% and 40% composites have 5 inputs whereas the ones built for the 60% and 80% have 7 inputs.

The number of neurons in the output layer of the neural networks is the same as the number of data points in the experimental loading-unloading curve for each composite. Table 5.3 shows the number of output neurons in each neural network model.

The number of hidden layers is fixed at two in line with the literature review conducted. (Chapter 3) In literature the number of neurons in the hidden layers are selected by a trial-and-error based approach. In this study, the number of neurons in each hidden layer is selected

Table 5.3 Number of neurons in the output layers of the neural network models developed for each composite

Composite	Number of neurons
20%	76
40%	70
60%	108
80%	82

depending on the number of neurons in the output layer of the neural networks. As a rule of thumb the number of neurons in the hidden layers is adjusted to be more than the number of neurons in the output layer. For the 20%, 40% and 80% composites this approach yielded successful results with 90, 80 and 90 neurons in each hidden layer respectively. Moreover, the learning rate and momentum term of the backpropagation algorithm are set to 0.4 and 0.6 respectively. These numbers are adopted through research conducted by H. Ceylan. [62]

The 60% composite is an especially challenging case due to its higher number of data points that are the result of additional loading/unloading cycles. For the 60% composite two sets of neural network models with 7-120-120-108 and 7-130-130-108 architectures are constructed. For the first set the learning rate and momentum term are set to 0.4 and 0.6 resepectively. For the second set they are set to 0.1 and 0.4. The neural network with the 7-130-130-108 architecture and 0.1 learning rate and 0.4 momentum term yielded the best results. Detailed analysis is presented at the results and discussion section.

5.4 Training and Testing of the Neural Networks

The back-propagation algorithm is used to train and test the neural networks. Presentation of a complete training and testing set to a neural network is called an epoch. At each epoch the dataset is randomized and the examples in the training set are used to modify the connection weights of the neural network. After the presentation of the examples in the training set is complete the connection weights of the neural network are fixed and the testing examples are presented to the neural network. The predictions obtained from the neural network are compared to FEM predictions to evaluate the performance of the neural network. This procedure

is repeated until a satisfactory match between the ANN and FEM predictions are obtained.

In the present study, outputs from the neural network models are elastic lattice strains in the fibers and total (macroscopic) strains in the composite. The ANN-predicted and given (FEM) strains for the fibers and the composite at the last testing set are plotted together and compared to each other to evaluate the neural network performance.

The data points on the ANN and FEM stress-strain curves are obtained by averaging the strains predicted by the neural network and the target strains calculated by the finite element model over 500 different input vectors presented at the last testing set. The residuals, R^2 values, of the two curves are calculated to quantitatively evaluate the accuracy of neural network predictions. The following formula is used to calculate the R^2 values:

$$R^2 = 1 - \left[\frac{\sum_i^n (FEM_i - ANN_i)^2}{\sum_i^n (FEM_i - FEM_{avg})^2} \right] \tag{5.2}$$

Here FEM_i denotes a strain value calculated by the finite element model at a data point on the loading-unloading curve, ANN_i denotes a strain value predicted by the forward neural network model at data point on the loading-unloading curve and FEM_{avg} denotes the average of n target strain values calculated by FEM either on the loading-unloading curve of the fibers or the composite. A "perfect" fit between ANN and FEM predictions would give an R^2 value of 1.

5.5 Results and Discussion

5.5.1 20% Composite

The results of the sensitivity study showed that θ_0 and θ_1 are insensitive for the 20% composite. Presence of insensitive parameters in neural network inputs adversely affects the neural networks ability in deciphering the functional relationship between the inputs and the outputs because, a change in an insensitive parameter does not yield any changes in the outputs. Therefore, the forward neural network models for the 20% composite are constructed based on the remaining 5 sensitive parameters. The same discussion applies to the 40% composite as well.

A 5-90-90-76 neural network is trained for 30,000 epochs by setting the learning rate and momentum term to 0.4 and 0.6 respectively. 3028 examples, i.e. FEM predictions, are used to train the neural network and 500 examples are used to test the neural network.

Figure 5.3 shows the comparison of FEM and ANN stress-strain curves for the 20% composite. The R^2 value for the fiber elastic strain and composite total strain are 0.999 for both curves.

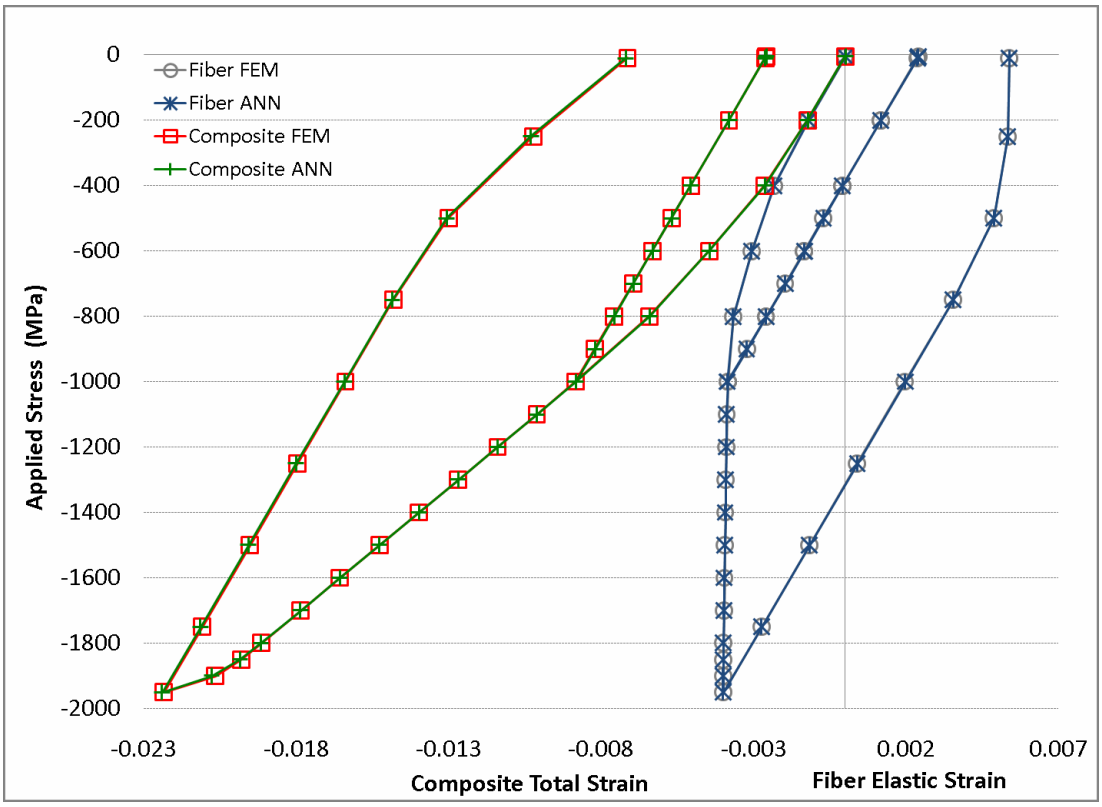


Figure 5.3 Predictions of the forward artificial neural network model for the 20 % composite

5.5.2 40% Composite

The results of the sensitivity study showed that θ_0 and θ_1 are insensitive for the 40% composite as well. Again, the forward neural network models for the 40% composite are constructed based on the remaining 5 sensitive parameters.

A 5-80-80-70 neural network is trained for 30,000 epochs by setting the learning rate and momentum term to 0.4 and 0.6 respectively. 4454 examples are used to train the neural network and 500 examples are used to test the neural network.

Figure 5.4 shows the comparison of FEM and ANN stress-strain curves for the 40 % composite. The R^2 value for the fiber elastic strain and composite total strain are 0.999 for both curves.

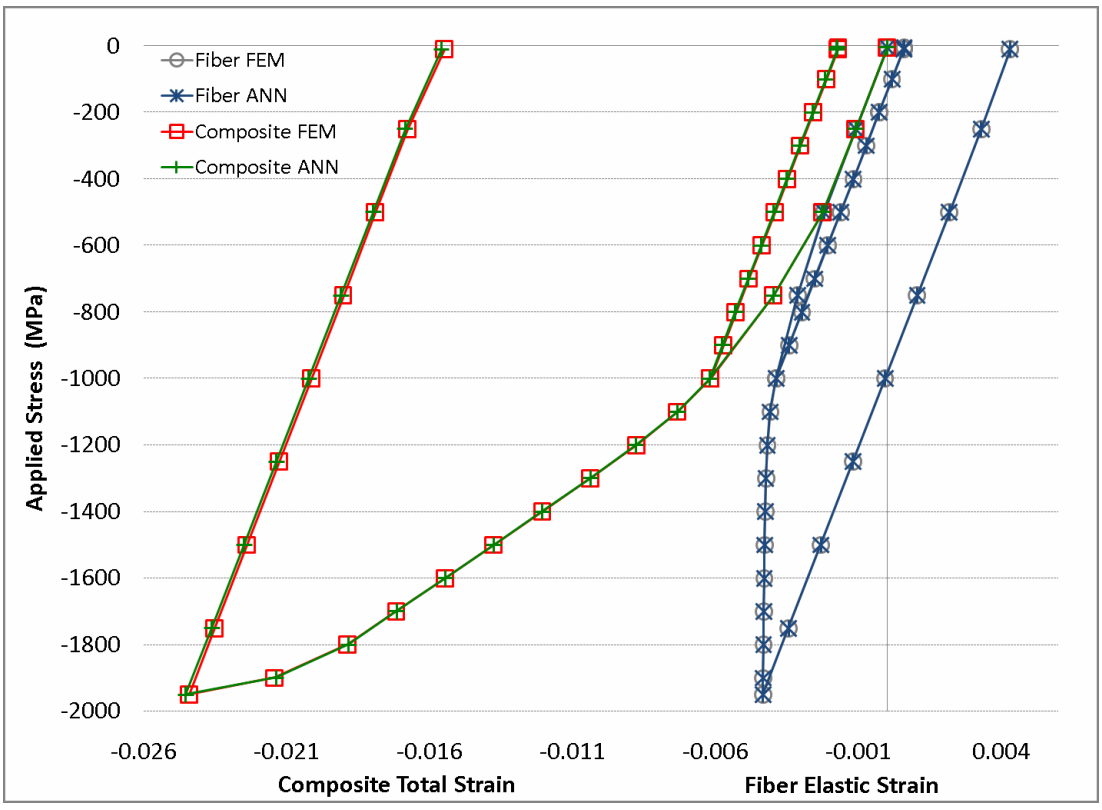


Figure 5.4 Predictions of the forward artificial neural network model for the 40 % composite

5.5.3 60% Composite

All of the seven parameters are sensitive for the 60% composite. The 60% composite is an especially challenging case due to its higher number of data points that are the result of additional loading/unloading cycles. For this case two sets of neural network models with 7-120-120-108 and 7-130-130-108 architectures are constructed. For the first set the learning rate and momentum term are set to 0.4 and 0.6 resepectively. For the second set they are set to 0.1 and 0.4. 3841 examples are used to train the neural networks and 500 examples are used to test the neural networks for 20,000 epochs.

Table 5.4 shows the R^2 values obtained for each case and figure 5.5 shows the comparison of the FEM and ANN predicted stress-strain curves for each case. These results show that the 7-130-130-108 (0.1-0.4) neural network gives the best match to the FEM.

Table 5.4 R^2 values obtained for the ANN-predicted fiber and composite stress-strain curves in comparison to the FEM-predicted stress-strain curve for the 60% composite.

Architecture	**Set**	**Fiber R^2**	**Composite R^2**	**Average R^2**
7-120-120-108	0.4-0.6	0.999955	0.999612	0.999612
7-120-120-108	0.1-0.4	0.999982	0.999454	0.999454
7-130-130-108	0.4-0.6	0.999931	0.999961	0.999561
7-130-130-108	0.1-0.4	0.999984	0.999720	0.999720

The 7-130-130-108 (0.1-0.4) neural network is trained for 20,000 more epochs to improve the match between the FEM and ANN curves even further. Figure 5.6 shows the comparison of final ANN predictions with the FEM predictions. The fiber R^2 value is calculated as 0.999978 while the composite R^2 value is calculated as 0.999842. This yields an average R^2 value of 0.999910. Despite the more complicated loading-unloading cycle, the forward ANN model matches the FEM model extremely well.

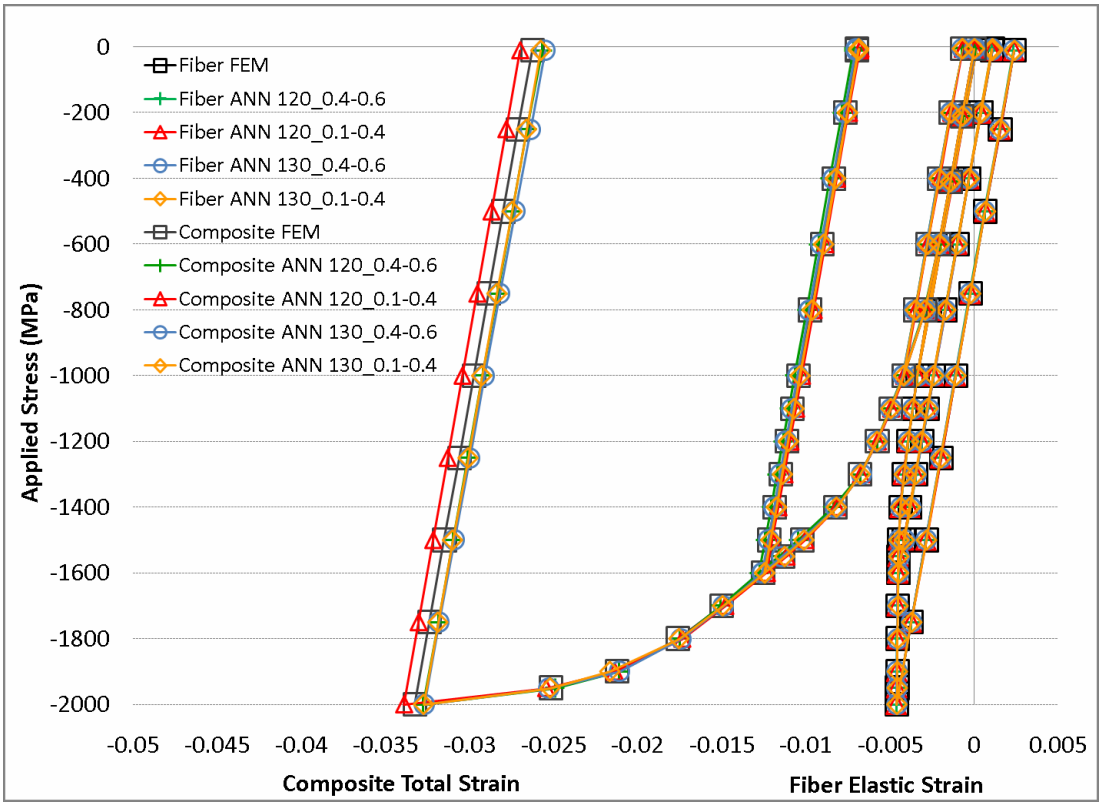


Figure 5.5 Comparison of ANN predicted stress-strain curves with the FEM predicted stress-strain curve for the four different cases studied for the 60% composite. The first number after ANN refers to the number of neurons in each hidden layer while the second and third numbers refer to the learning rate and momentum term respectively.

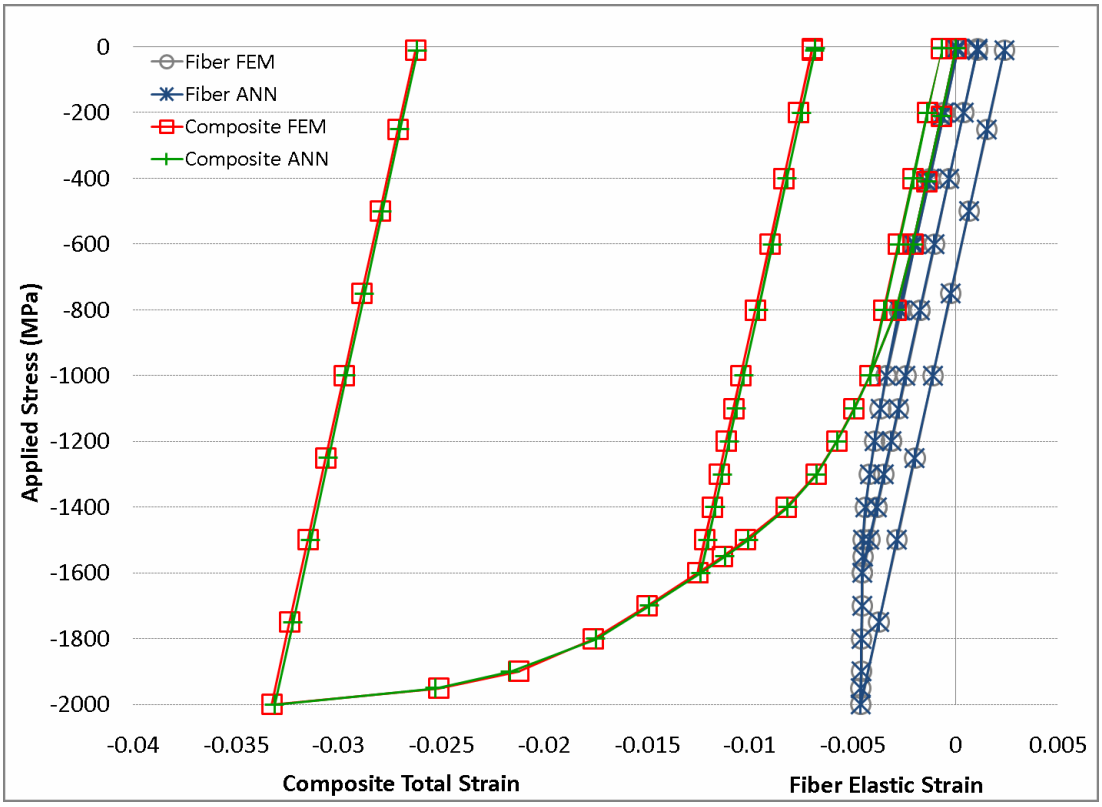


Figure 5.6 Predictions of the forward artificial neural network model for the 60 % composite

5.5.4 80% Composite

All of the seven parameters are sensitive for the 80% composite. A 7-90-90-82 neural network is trained for 20,000 epochs by setting the learning rate and the monetum term to 0.4 and 0.6. 4374 examples are used to train the neural network and 500 examples are used to test it.

Figure 5.7 shows the comparison of FEM and ANN stress-strain curves for the 80 % composite. The R^2 value for the fiber elastic strain and composite total strain are 0.999 for both curves.

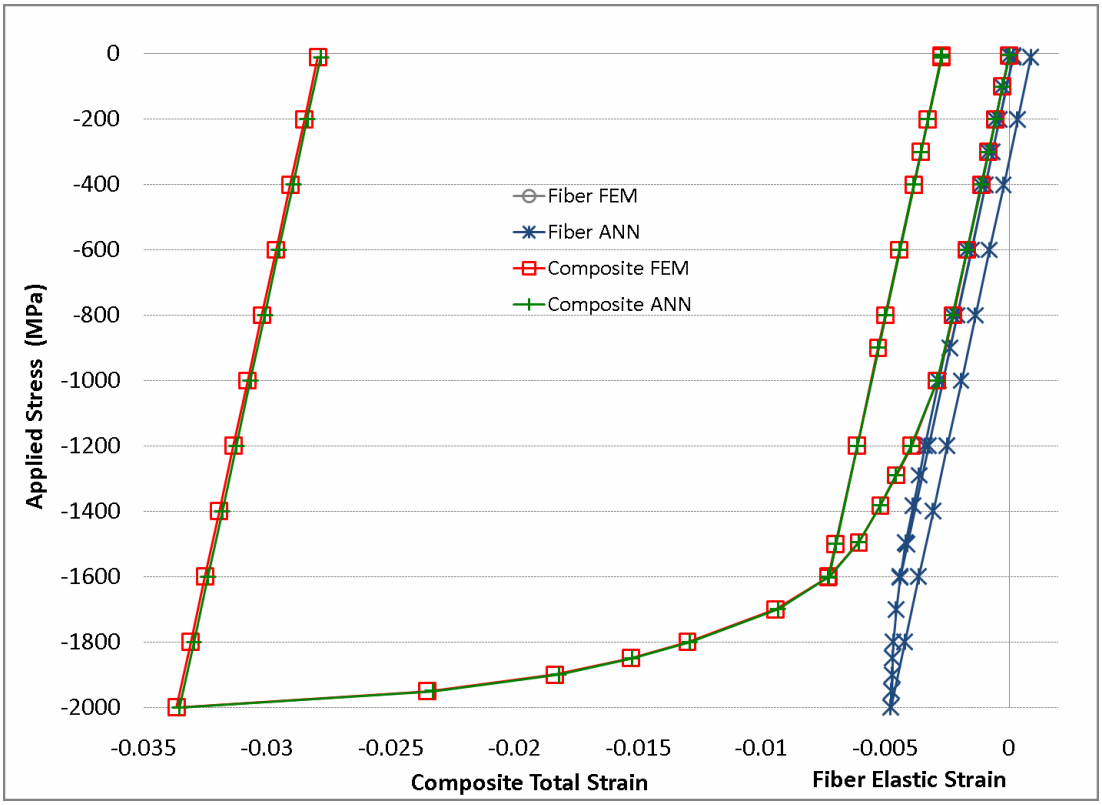


Figure 5.7 Predictions of the forward artificial neural Network Model for the 80 % Composite

5.6 Conclusions

All forward neural network models successfully predicted the stress-strain relationship estimated by the FEM. This means, these forward ANN models can easily substitute for the FEM (for the current set of material parameters and morphologies) yielding results at a small fraction of the time required to run the FEM. The results of the sensitivity study played a crucial role in designing the neural network models, for instance, by reducing the complexity of the ANN model when insensitive input (material) parameters are excluded. The forward neural network models developed here are used to refine the inverse analysis results as described in chapter 7.

CHAPTER 6. INVERSE NEURAL NETWORK MODELS

6.1 Introduction

In the present study, the aim of an inverse neural network model is to predict the optimized value of a certain parameter governing the constitutive behavior of the composites. For each composite optimized values of the sensitive parameters are predicted one by one, then these values are used to plot stress-strain curves by using FEM and the forward neural network models developed. Finally, these stress-strain curves are compared to each other and to the experimental stress-strain curves to validate the predictions of the inverse neural network models.

The inputs to the inverse neural network models are strains at the data points on the loading-unloading curves for the fiber elastic strain and the composite total strain. Figure 6.1 shows a schematic representation of the inverse neural network models on one cartoon. Note that each inverse neural network model actually has only one output.

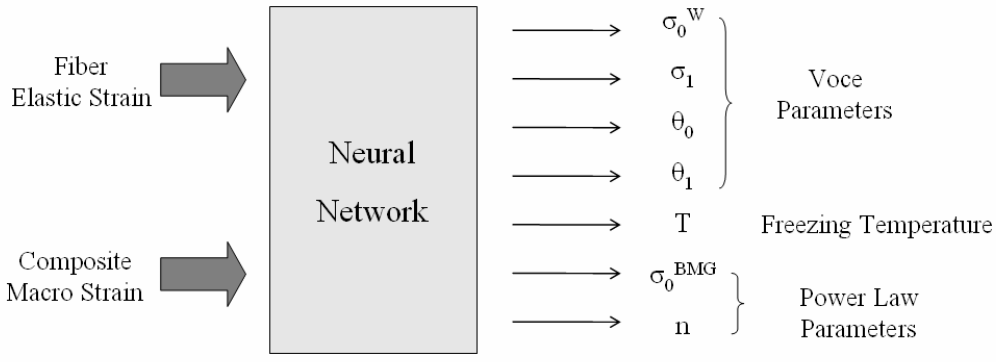


Figure 6.1 Schematic representation of the inverse neural network model

In inverse neural network models each of the output parameters has a different functional relationship with the input parameters and therefore, constructing only one neural network to predict all output parameters does not give satisfactory results. For forward neural network models it is a simpler task to predict two groups of dependent parameters from seven independent input parameters. However, for the inverse neural network model it is more complicated to predict seven independent parameters from two groups of dependent parameters. Consequently, a separate inverse neural network model is constructed to predict each output parameter.

To be able to obtain the best results four inverse neural network models with different architectures are constructed for each sensitive parameter of each composite. Then, the architecture yielding the lowest average mean squared error at convergence for the training and testing sets is selected for each composite. For the selected architectures the neural network training is repeated with two different arrays of initial connection weights. Finally, three stress-strain curves are plotted for each composite and compared to the experimental results.

6.2 Generation of Training and Testing Sets

The database generated by finite element analysis for the forward neural network models is also used for training and testing the inverse models. However, there is a significant difference in presentation of the data to the neural network and normalization. In this case, the inputs to the neural network are the strains at the data points on the stress-strain curves for the fiber elastic strain and the composite total strain and, the outputs are the material parameters and the freezing temperature. In other words, the inputs to the forward model become the outputs of the inverse model and the outputs of the forward model become the inputs to the inverse model. The strains for the fibers and the composite are treated as two separate groups and normalized from their ranges to [-2.5,2.5]. Each of the seven output parameters are normalized from their individual ranges to [0.1,0.9]. Details of normalization are presented in Chapter 5.

6.3 Training of the Inverse Neural Network Models

The back-propagation algorithm is used to train and test the neural networks. The learning rate and the momentum term of the back-propagation algorithm is tailored for individual neural network models. The initial values of these parameters are adopted through research conducted by H. Ceylan. [62] The learning rate is set to 0.4 and decreased when oscillation of neural network predictions is observed. It is found that setting the learning rate below 0.1 slows down neural network training drastically and should be avoided until and unless it is necessary. The momentum term is set to 0.6 and modified when needed. Decreasing the momentum term also improves oscillation. However, the main function of the momentum term is to prevent the neural network from getting stuck in a local minimum. It should be noted that increasing the momentum term above 0.9 may cause instability of the neural network. Consequently, special attention must be paid to remedy this situation. Increasing the learning rate and the momentum term speeds up neural network training. However, in adjusting these two parameters there is always a compromise between the prediction performance and the training time. Moreover, the optimum values of them depend on the problem studied and the neural network architecture. In this study, best results are obtained by varying the learning rate and the momentum term in the ranges [0.1,0.4] and [0.4,0.6] respectively.

The training of the neural networks are continued until the neural network converges or the mean squared error (MSE) in neural network predictions falls below a predetermined value. Here convergence of the neural network refers to the point where further training does not yield improvement in the predictions. The mean squared error is given by the following formula:

$$MSE = \left[\frac{\sum_{i=1}^{N} (FEM_i - ANN_i)^2}{N} \right] \tag{6.1}$$

where FEM_i denotes a target value, ANN_i denotes a neural network prediction and N denotes the number of examples in the training or testing set.

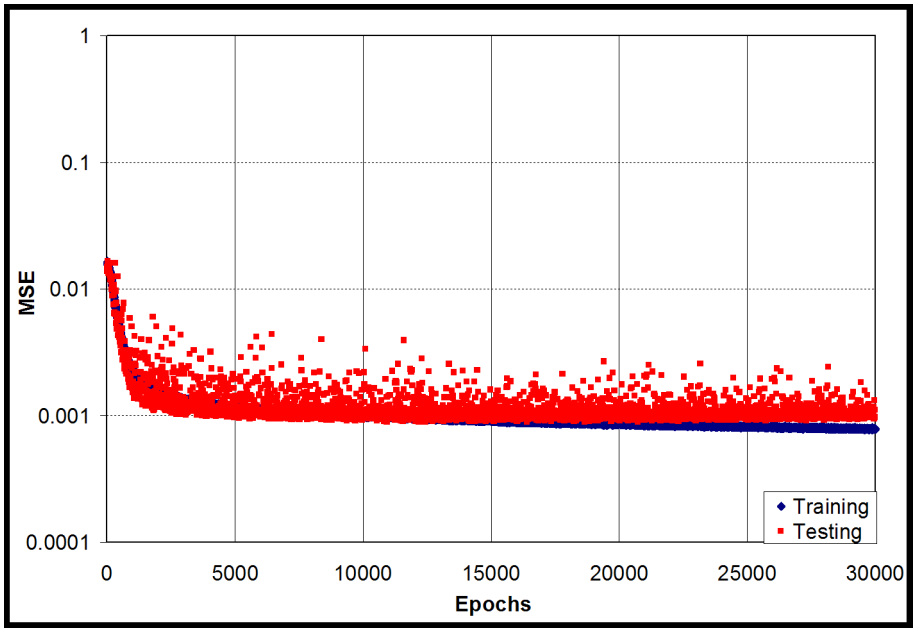


Figure 6.2 Sample learning curve

The progress of neural network training is monitored by a *learning curve*. Figure 6.2 shows a sample learning curve. A learning curve shows the change in MSE for the training and testing sets throughout the training session. At the end of each epoch (training-testing cycle) the MSE for the training and the testing sets are calculated and plotted versus number of epochs elapsed. This way the rate of learning and the convergence behavior of the neural network can be graphically monitored. Generally, it starts with a steep decrease in the training and testing errors and then gradually reaches a plateau where the neural network is said to be converged. For an insensitive parameter a neural network would not converge; consequently, neural network training is not performed for insensitive parameters.

6.4 Testing of the Inverse Neural Network Models

After a neural network is converged, the predictions are tested by a *line of equality plot*. The x-axis of this plot represents the given target values and the y-axis represents the corresponding neural network predictions. Each point on the plot represents a target-output pair in the testing set at the final epoch of a training session. When these two values match, the point appears on

the x=y line (line of equality). This way the predictions and targets are qualitatively compared on a single plot. The closer the points are on the plot to the line of equality, the more accurate are the neural network predictions. Figure 6.3 shows an example line of equality plot.

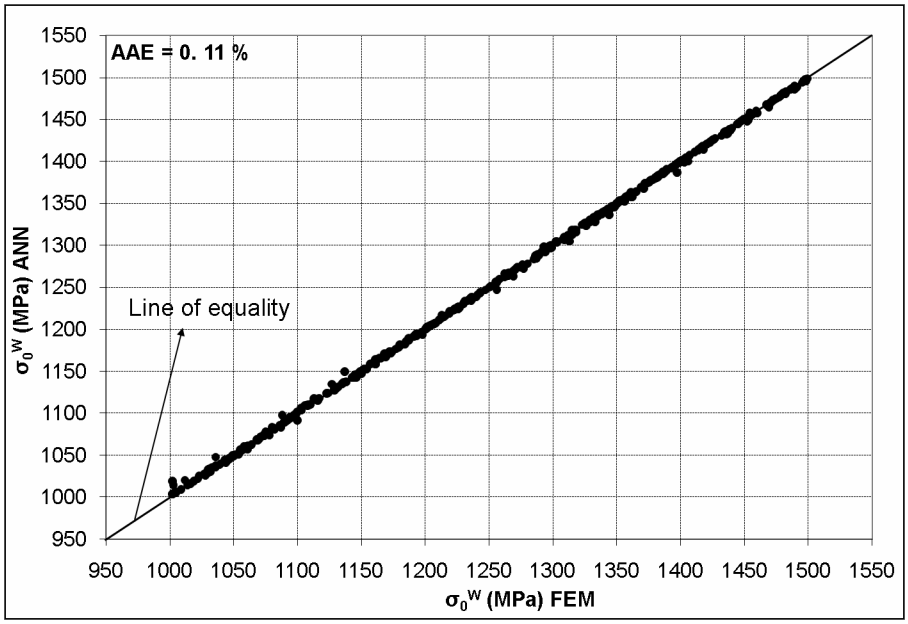


Figure 6.3 Sample line of equality plot (σ_0^W)

Percentage average absolute error between the predictions and the targets are calculated to enable quantitative evaluation of the neural network predictions. The percentage average absolute error (%AAE) is calculated by the following formula:

$$\%AAE = \left[\frac{\sum_{i=1}^{N} \left(\frac{|FEM_i - ANN_i|}{FEM_i} \times 100 \right)}{N} \right] \tag{6.2}$$

where FEM_i denotes a target value, ANN_i denotes a neural network prediction and N denotes the number of examples in the testing set. The $\%AAE$ cannot be used to evaluate the neural network performance alone but it is a complementary measure to the qualitative nature of the line of equality plot.

Mean squared errors presented on the learning curves are calculated based on the normalized values of the targets and the predictions in the range [0.1,0.9]. Percentage average absolute errors are calculated based on the unnormalized values of the targets and the predictions are displayed on the line of equality plots. The unnormalized ranges are parameter specific and can be read from the line of equality plots.

6.5 Neural Network Architectures

Four different inverse neural network architectures are investigated for each composite. The number of nodes in the input layer is the same as the number of data points on the experimental loading-unloading curves for each composite. All neural networks have one output and two hidden layers. The number of neurons in each hidden layer is varied depending on the number of nodes in the input layer.

The number of neural network models developed for each composite depends on the number of sensitive output parameters affecting the constitutive behavior of the composite. θ_0 and θ_1 are found to be insensitive for the 20% and 40% composites as described in chapter 4. Therefore, 20 inverse neural network models – 5 per achitecture, are developed for the the 20% and 40% composites whereas 28 inverse neural network models – 7 per architecture, are developed for the 60% and 80% composites. It sums to 96 inverse neural network models total.

6.5.1 20% Composite

76-10-5-1, 76-50-25-1, 76-100-50-1 and 76-150-75-1 neural network architectures are investigated for the 20% composite. The total number of neurons in the hidden layers is varied between a minimum and a maximum depending on the number of nodes in the input layer.

σ_0^W, σ_1, freezing temperature, σ_0^{BMG} and n are the sensitive parameters for this case. Five separate neural networks are trained per architecture to predict each parameter. The neural network trainings are continued until convergence and the average mean squared training and testing errors are calculated for each neural network for the last 250 epochs of training. This way the average training and testing errors at convergence are calculated for each neural

network and used as the architecture selection criteria.

The average convergence errors for the training and testing sets of the 20% composite are shown on Figure 6.4 and 6.5 respectively. Note that, changing the neural network architecture does not yield a noticeable difference in the average training and testing errors obtained for either σ_0^{BMG} or n. This indicates that these two parameters are not very sensitive, which is already known from the results of the sensitivity study. Although limited improvement is achieved in the freezing temperature predictions, the average convergence training and testing errors are very low which, in turn yields high prediction accuracy. On the other hand, the training and testing errors decrease significantly for σ_0^W and σ_1. The 76-150-75-1 neural network yields higher testing errors at convergence for σ_1 and freezing temperature when compared to the 76-100-50-1 neural network. Moreover, adding 75 additional neurons to the architecture slows down the training process drastically. Therefore, the 76-100-50-1 neural network yielded the best performance among the four architectures investigated.

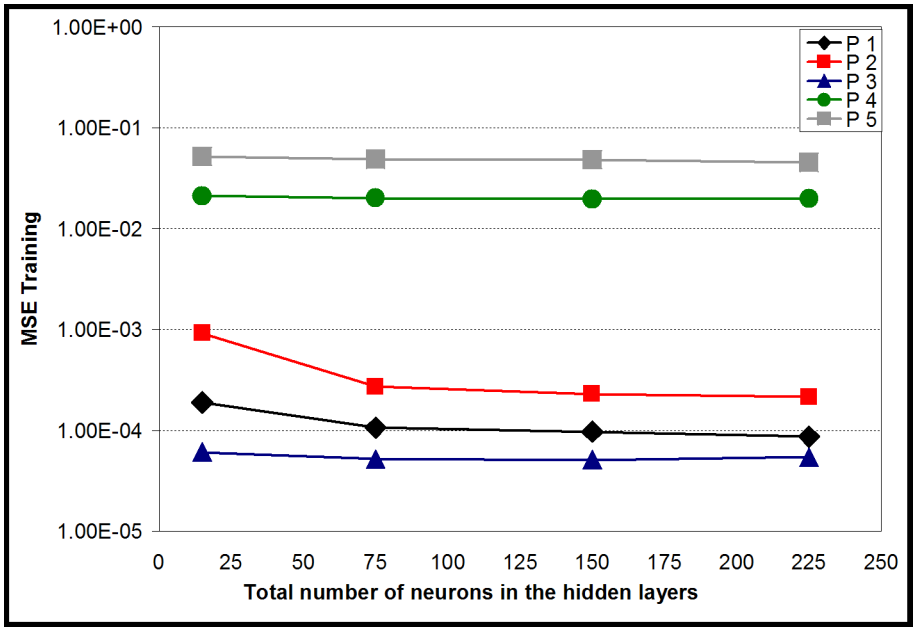


Figure 6.4 Comparison of convergence training errors obtained for the 20% composite. P1 - σ_0^W, P2 - σ_1, P3 - freezing temperature, P4 - σ_0^{BMG}, P5 - n.

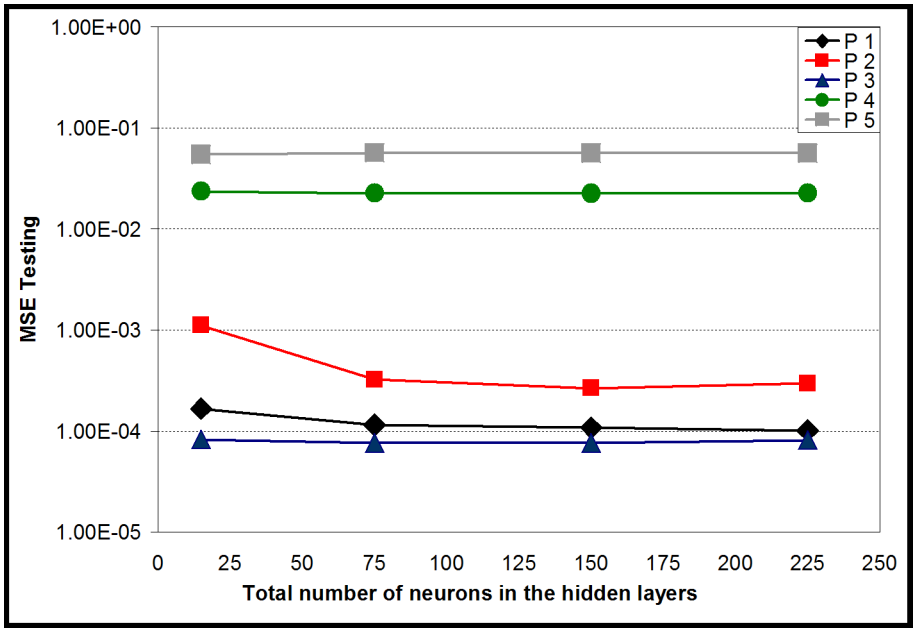


Figure 6.5 Comparison of convergence testing errors obtained for the 20% composite. P1 - σ_0^W, P2 - σ_1, P3 - freezing temperature, P4 - σ_0^{BMG}, P5 - n.

6.5.2 40% Composite

70-10-5-1, 70-50-25-1, 70-100-50-1 and 70-150-75-1 neural network architectures are investigated for the 40% composite. σ_0^W, σ_1, freezing temperature, σ_0^{BMG} and n are the sensitive parameters for this case as well. Five separate neural networks are trained per architecture and, the average convergence training and testing errors are caluculated and presented on Figure 6.6 and 6.7 respectively.

In this case, very limited improvement is obtained in the average convergence training and testing errors for the freezing temperature, σ_0^{BMG} and n. Moreover, the calculated errors for the freezing temperature are significantly higher than the ones obtained for the 20% composite which, in turn affects the prediction accuracy for the parameter negatively. On the other hand, significant improvement is observed in the predictions for σ_0^W and σ_1. The 70-100-50-1 neural network gives the best combination of prediction accuracy and training time for the 40% composite.

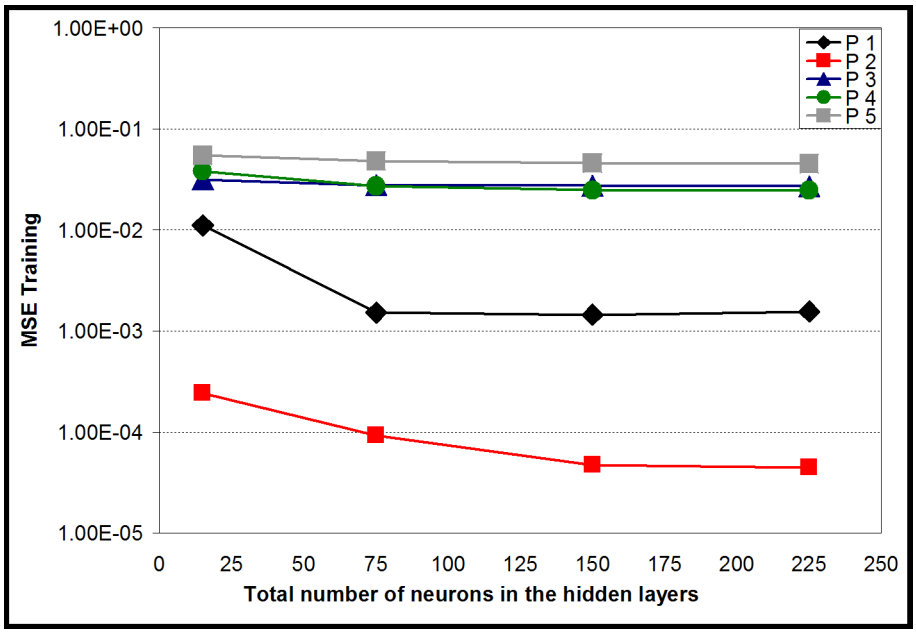


Figure 6.6 Comparison of convergence training errors obtained for the 40% composite. P1 - σ_0^W, P2 - σ_1, P3 - freezing temperature, P4 - σ_0^{BMG}, P5 - n.

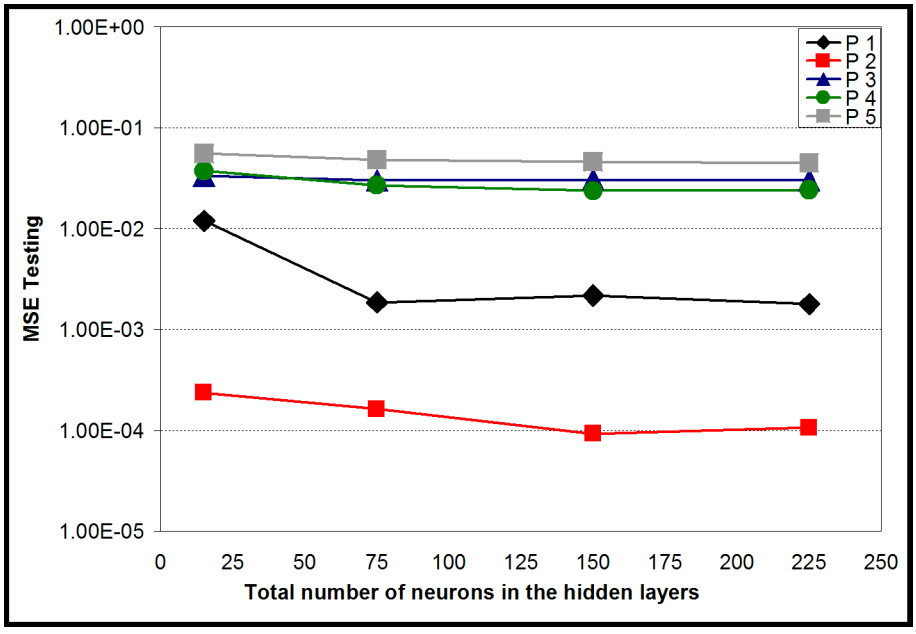


Figure 6.7 Comparison of convergence testing errors obtained for the 40% composite. P1 - σ_0^W, P2 - σ_1, P3 - freezing temperature, P4 - σ_0^{BMG}, P5 - n.

6.5.3 60% Composite

108-10-5-1, 108-100-50-1, 108-120-60-1 and 108-140-70-1 neural network architectures are investigated for the 60% composite. Due to the higher number of data points on the experimental stress-strain curve and the more complex loading-unloading cycle, more neurons are employed in these neural network architectures.

Although all seven parameters are found to be sensitive to some extent in the sensitivity study, only σ_1 showed considerable improvement in the average convergence training and testing errors upon a change in neural network architecture. Despite the limited improvement observed for σ_0^W, the errors obtained are low enough to give accurate predictions. Figure 6.8 and 6.9 shows the calculated errors for the training and testing sets respectively. The 108-140-70-1 neural network yielded the lowest errors for the 60% composite.

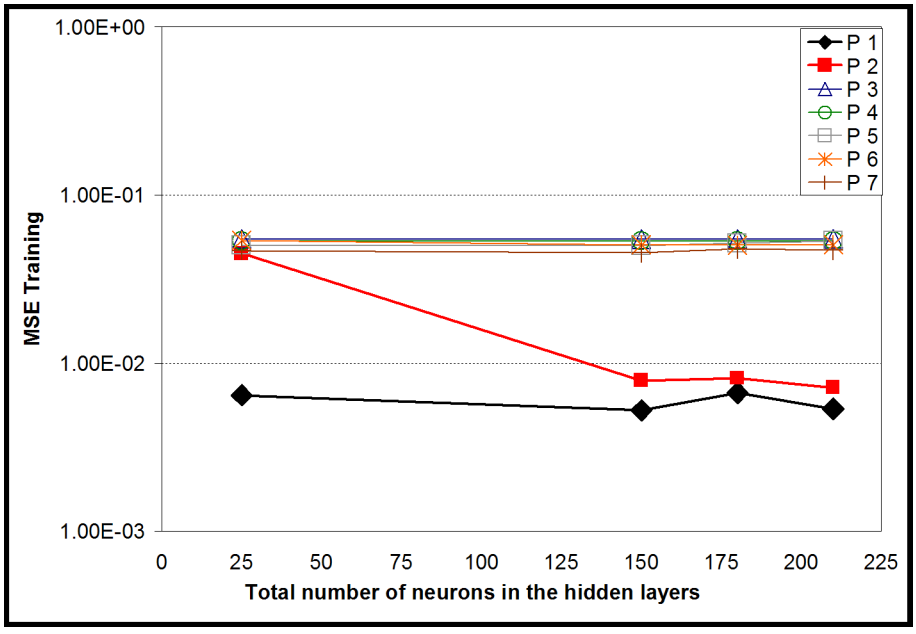


Figure 6.8 Comparison of convergence training errors obtained for the 60% composite. P1 - σ_0^W, P2 - σ_1, P3 - freezing temperature, P4 - σ_0^{BMG}, P5 - n.

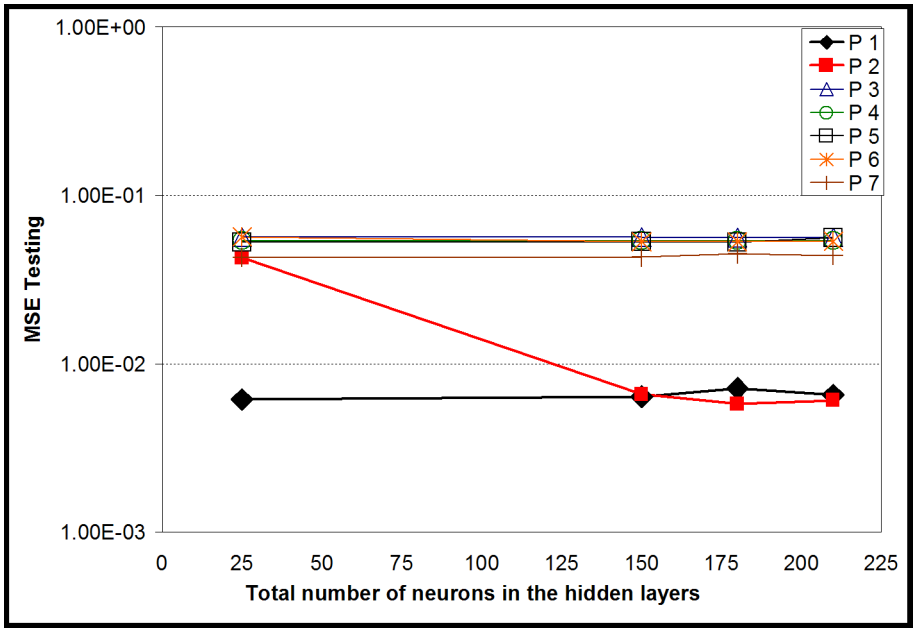


Figure 6.9 Comparison of convergence testing errors obtained for the 60% composite. P1 - σ_0^W, P2 - σ_1, P3 - freezing temperature, P4 - σ_0^{BMG}, P5 - n.

6.5.4 80% Composite

82-10-5-1, 82-60-30-1, 82-120-60-1 and 82-150-90-1 neural network architectures are investigated for the 80% composite. Although all seven parameters are found to be sensitive to some extent in the sensitivity study, only σ_1 showed considerable improvement in the average convergence training and testing errors similar to the 60% case. Moreover, the improvement achieved for σ_0^W is limited but the predictions obtained are satisfactory.

Figure 6.10 and 6.11 shows the calculated errors for the training and testing sets respectively. The 82-150-90-1 neural network yielded higher training and testing errors at convergence for σ_1 when compared to the 82-120-60-1 neural network. Therefore, the 82-120-60-1 neural network is selected as the best architecture for the 80% composite.

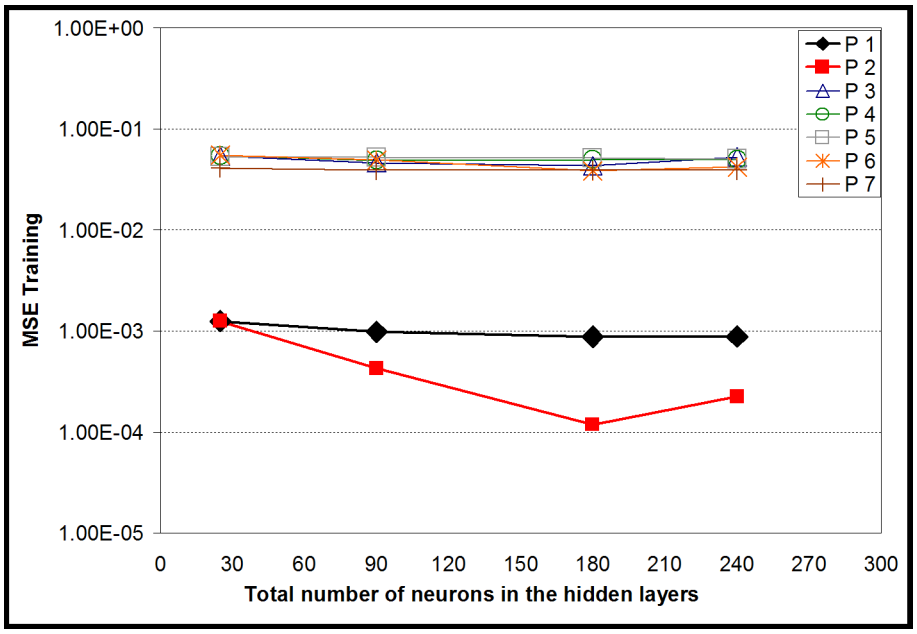


Figure 6.10 Comparison of convergence training errors obtained for the 80% composite. P1 - σ_0^W, P2 - σ_1, P3 - freezing temperature, P4 - σ_0^{BMG}, P5 - n.

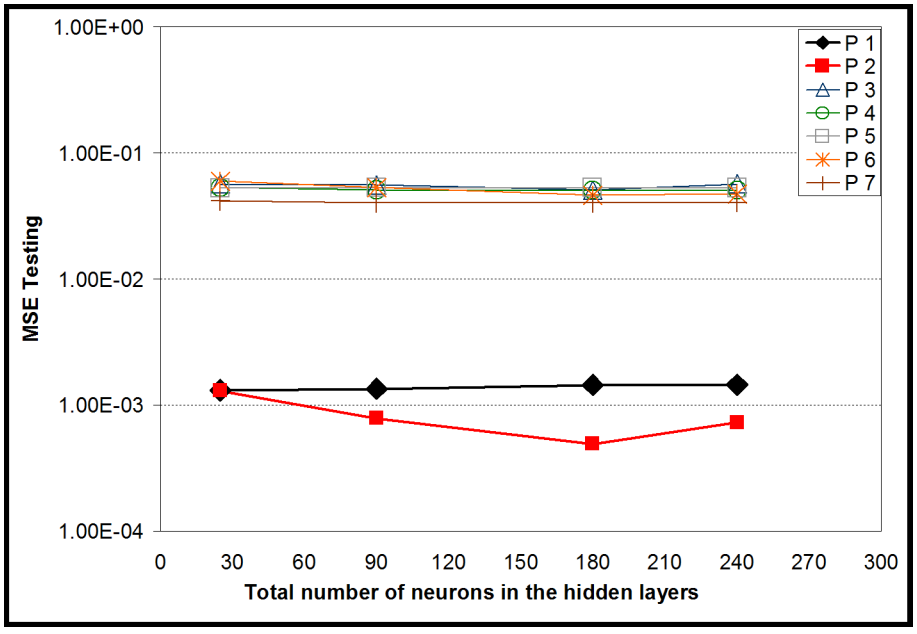


Figure 6.11 Comparison of convergence testing errors obtained for the 80% composite. P1 - σ_0^W, P2 - σ_1, P3 - freezing temperature, P4 - σ_0^{BMG}, P5 - n.

6.6 Validation of the Inverse Neural Network Models

Table 6.1 shows the neural network architectures selected for each composite. The inverse neural network trainings are continued with the selected architectures by training two more models for each composite with different initial array of connection weights. The connection weights are obtained from H. Ceylan. The new neural network models with the same architecture but different initial array of connection weights are referred to as Set 1, Set 2 and Set 3.

Table 6.1 Selected neural network architectures

Composite	**Architecture**
20%	76-100-50-1
40%	70-100-50-1
60%	108-140-70-1
80%	82-120-60-1

The predictions from the three sets of neural network models are presented and used to plot the fiber elastic strain and composite total strain loading-unloading curves for each composite by using FEM and the forward neural network models developed in Chapter 5. The stress-strain curves obtained from FEM and the forward models are compared to each other and to the experimental stress-strain curves for validating the neural network framework developed. The experimental curves are plotted for the fiber elastic strain and composite total strain by using data obtained by neutron diffraction and simultaneous macroscopic strain measurements obtained by an extensometer respectively.

The inverse neural network predictions are obtained by presenting the experimental stress-strain curves of each composite to the trained inverse neural network models. Consequently, each prediction presented on the tables is obtained from an inverse neural network specifically trained for that parameter. The predictions are presented along with the percentage average absolute errors (%AAE, Equation 6.2) obtained after the testing of the neural network models. Here, the aim of presenting the %AAE is supplying a measure of confidence in the inverse model predictions.

6.6.1 20% Composite

Table 6.2 Inverse neural network predictions for the 20% composite

Set	σ_0^W **(MPa)**	σ_1 **(MPa)**	**T (°C)**	σ_0^{BMG} **(MPa)**	**n**
1	-938	-828	278	-2102	11
% AAE	0.15	0.30	0.29	1.35	54
2	-938	-825	318	-2105	8
% AAE	0.11	0.29	0.29	1.37	53
3	-938	-835	280	-2089	7
% AAE	0.11	0.28	0.29	1.40	54

Table 6.2 shows the predictions obtained from the 3 sets of inverse neural networks trained for the 20% composite and Table 5.2 shows the ranges of the seven parameters employed in the FEM in gererating the training and testing sets for the neural network models. Note that, the inverse neural network models made predictions outside the training ranges for σ_0^W, σ_1 and σ_0^{BMG}. The reasons for this counter-intuitive result is explained in the conclusions in detail.

The %AAE values obtained for σ_0^W, σ_1 and initial temperature are very low and the neural network predictions are consistent. Generally a %AAE value smaller than 0.5% indicates a very good training. The %AAE value obtained for σ_0^{BMG} is also fairly good, however the 53-54 %AAE obtained for the power law exponent (n) is very high and the neural network predictions are inconsistent. Note that, the accuracy of each prediction affects the overall stress-strain curve proportional to the parameter's sensitivity and the match between the experimental and the simulation curves are mostly controlled by the most sensitive parameters.

θ_0 and θ_1 are assigned to their mean values and presented to the FEM along with the inverse neural network predictions to be able to plot the FEM stress-strain curves shown in Figures 6.12 through 6.14. As the forward neural network model for the 20% composite is constructed for 5 input parameters, the inverse neural network predictions are directly presented to the forward model to plot the ANN curves. The same discussion applies to the 40% case as well. (Figures 6.15 through 6.17)

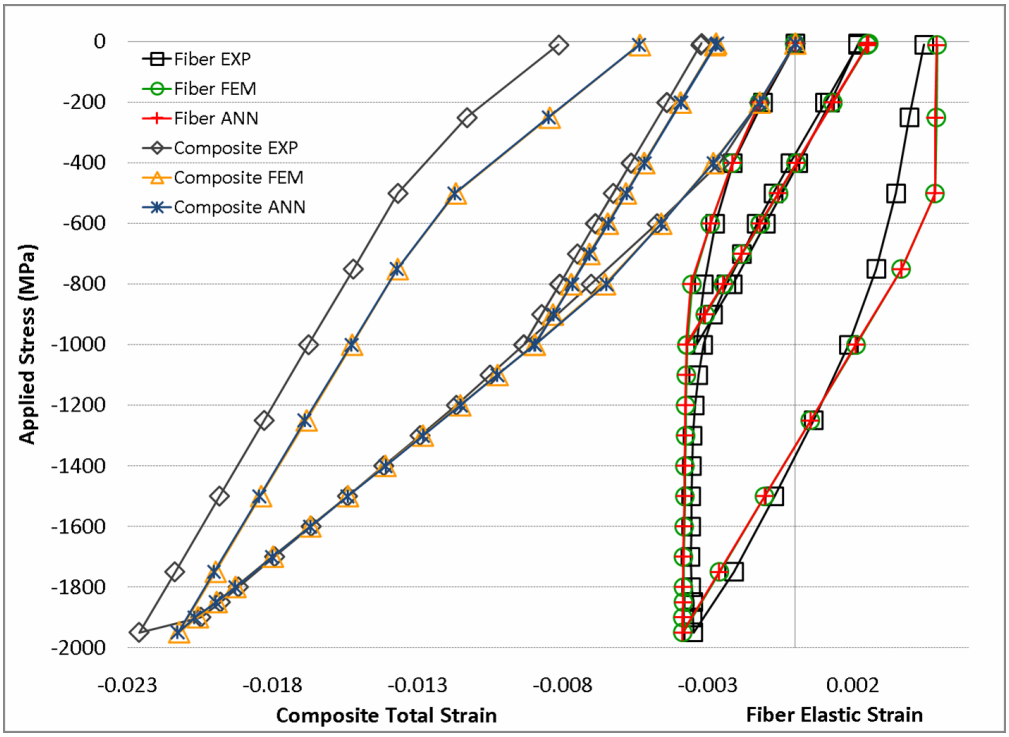


Figure 6.12 Comparison of the experimental, FEM and ANN stress-strain curves for the 20% composite. (Set 1)

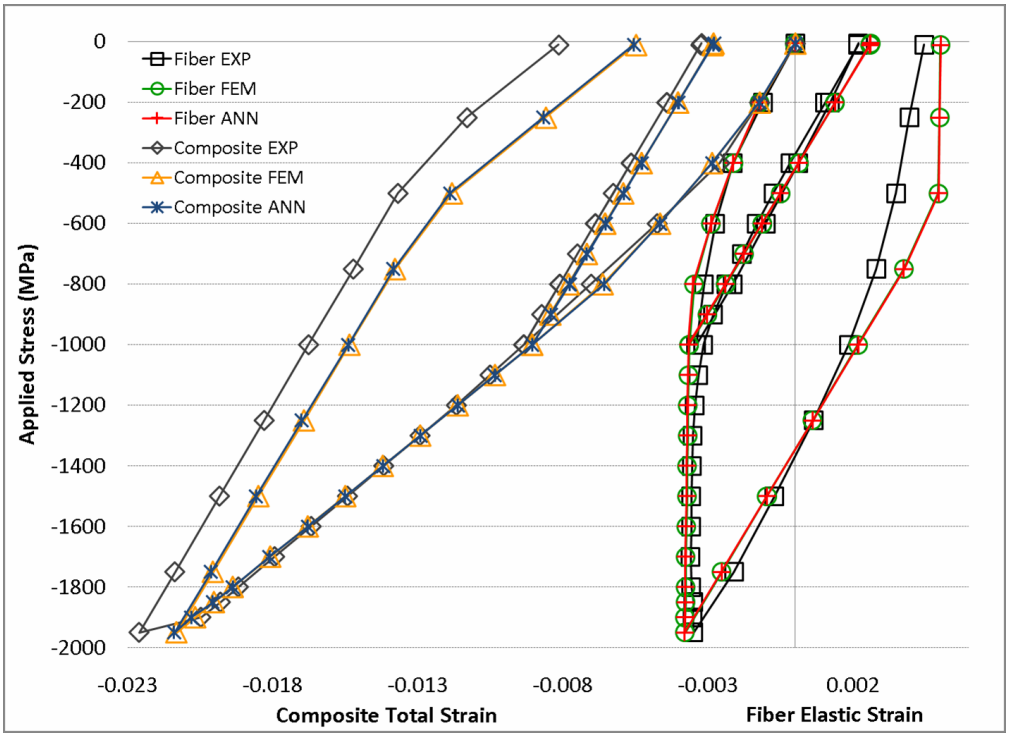


Figure 6.13 Comparison of the experimental, FEM and ANN stress-strain curves for the 20% composite. (Set 2)

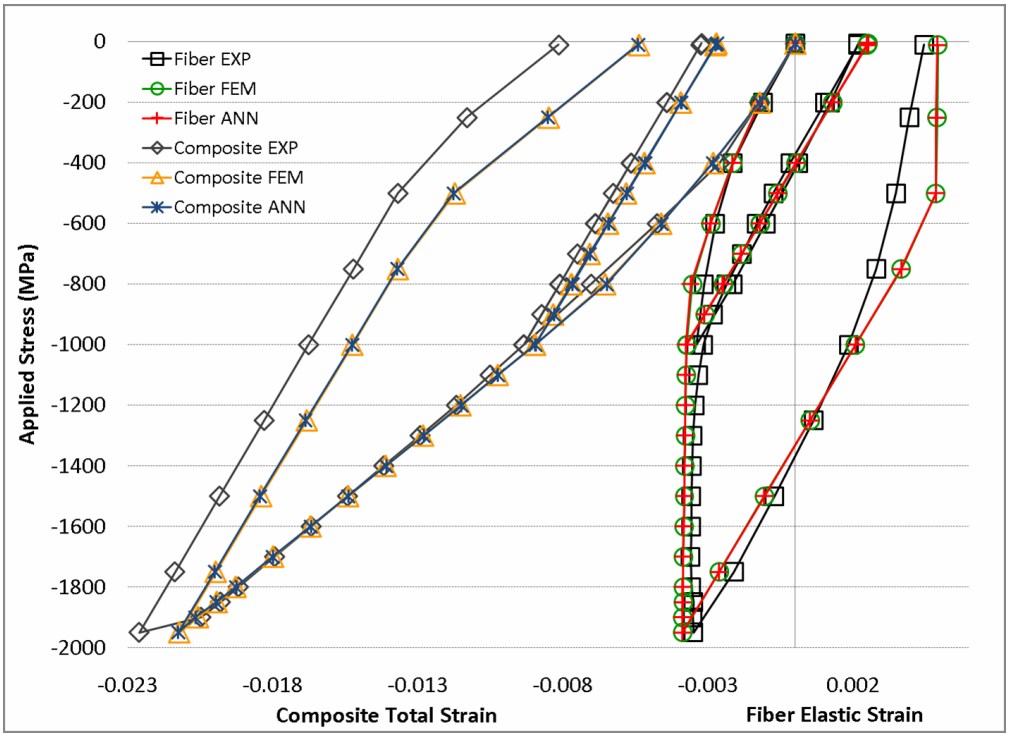


Figure 6.14 Comparison of the experimental, FEM and ANN stress-strain curves for the 20% composite. (Set 3)

The residuals (R^2 values, Equation 5.2) between the FEM and the ANN stress-strain curves are 0.999 for all sets which, in turn confirms the accuracy of forward model predictions. Table 6.3 shows the residuals obtained between the experimental and the ANN curves. It is found that predictions from Set 2 yields the best match between the experimental data and the neural network predictions. However, there is a considerable mismatch between the experimental stress-strain curves and the best predictions obtained from the inverse neural network models. This can be explained by the "out-of-range" predictions made by the inverse neural network models. The inverse neural network predictions for the 20% composite are further refined by an iterative optimization technique, which is explained in the next chapter.

Table 6.3 R^2 (residual) values between the experimental and the ANN curves for the 20% composite

Set	**Fiber R^2**	**Composite R^2**	**Average R^2**
1	0.972	0.975	0.9735
2	0.971	0.979	0.9750
3	0.971	0.975	0.9730

6.6.2 40% Composite

Table 6.4 Inverse neural network predictions for the 40% composite

Set	σ_0^W **(MPa)**	σ_1 **(MPa)**	**T (°C)**	σ_0^{BMG} **(MPa)**	**n**
1	-1558	-639	205	-2037	9
% AAE	0.26	0.21	11.5	1.35	41
2	-1561	-560	316	-1965	6
% AAE	0.28	0.21	11.2	1.32	41
3	-1561	-836	300	-1969	5
% AAE	0.26	0.23	11.2	1.60	41

Table 6.4 shows the predictions obtained from the 3 sets of inverse neural networks trained for the 40% composite. Note that, the predictions for σ_0^W are "out-of-range". (Table 5.2) Moreover, there is a large variation in the predictions obtained for σ_1.

The %AAE values obtained for σ_0^W and σ_1 are very low however, the predictions for σ_1 are inconsistent. Note that, the % AAE value for the initial temperature is significantly higher than that of the 20% composite. This outcome was expected because the results obtained in the previous section showed that initial temperature became insensitive to the changes in the neual network architecture for the 40%, 60% and 80% composites and the average convergence errors increased considerably.

Figures 6.15 through 6.17 show the comparison of the experimental, FEM and ANN stress-strain curves and, Table 6.5 shows the residuals calculated between the experimental and the ANN curves for the 40% composite. There is significant variation in the residuals obtained for each set, due to the variation in the σ_1 predictions. The residuals between the ANN and FEM curves are 0.999 except for the composite total strain curve of set 3, which has a residual of 0.997.

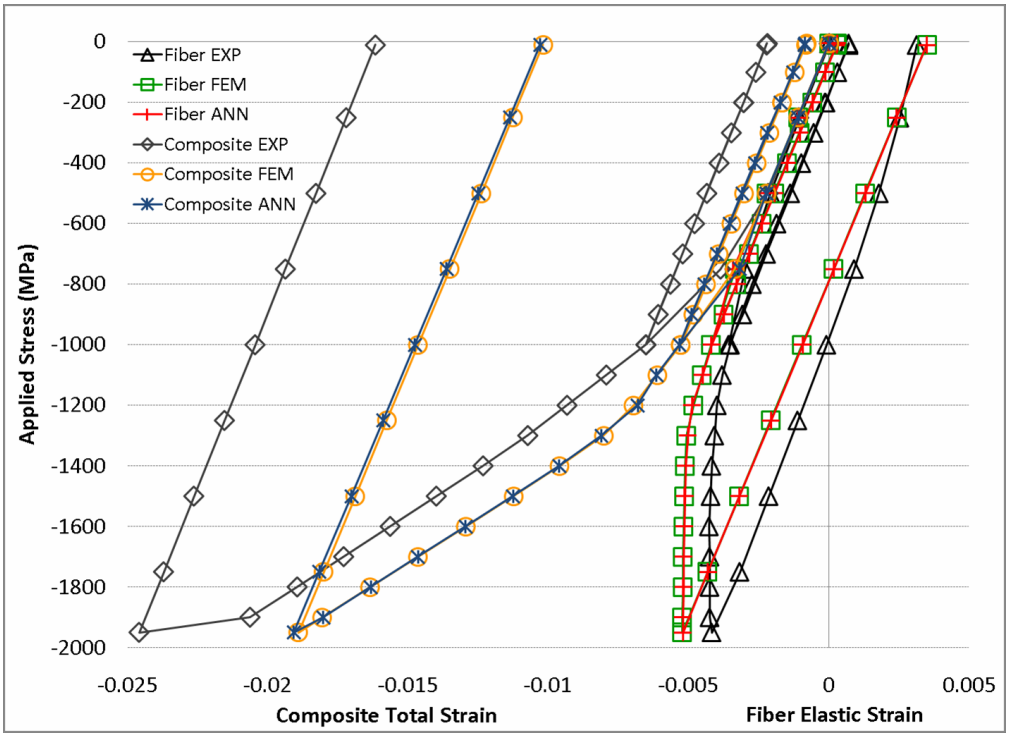


Figure 6.15 Comparison of the experimental, FEM and ANN stress-strain curves for the 40% composite. (Set 1)

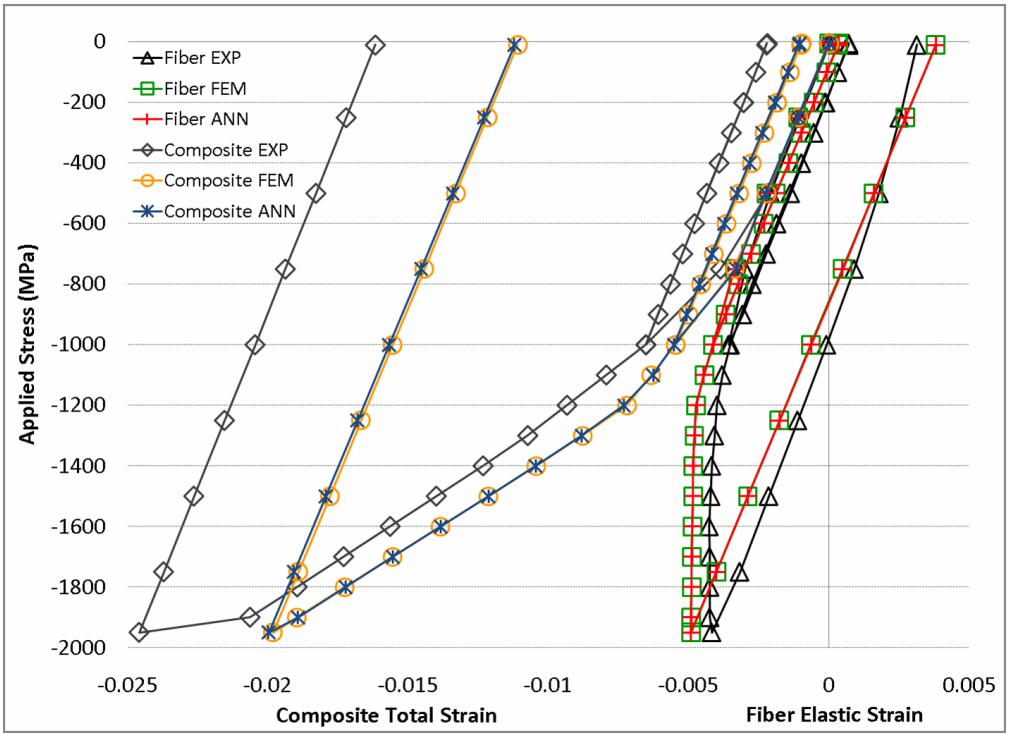


Figure 6.16 Comparison of the experimental, FEM and ANN stress-strain curves for the 40% composite. (Set 2)

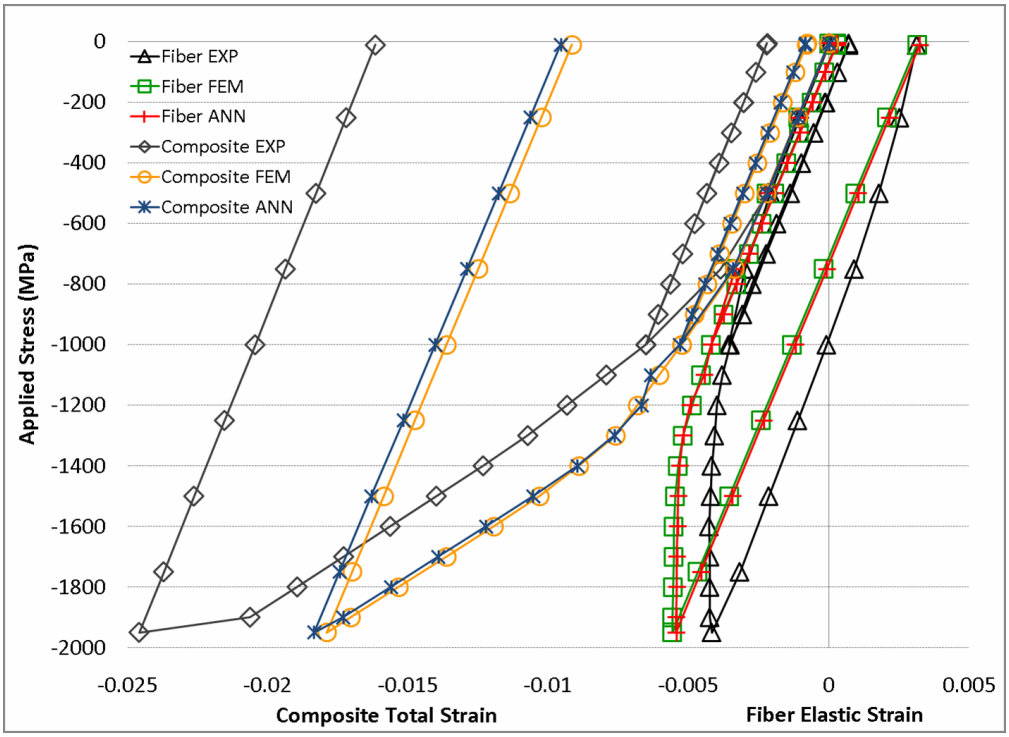


Figure 6.17 Comparison of the experimental, FEM and ANN stress-strain curves for the 40% composite. (Set 3)

Set 2 yields the highest residual for the 40% composite. However, there is a significant mismatch between the ANN and the experimental stress-strain curves. Therefore, the inverse neural network predictions are refined by iterative optimization and the results are presented in the next chapter.

Table 6.5 R^2 (residual) values between the experimental and the ANN curves for the 40% composite

Set	**Fiber R^2**	**Composite R^2**	**Average R^2**
1	0.892	0.814	0.8530
2	0.971	0.874	0.9225
3	0.825	0.760	0.7925

6.6.3 60% Composite

Table 6.6 Inverse neural network predictions for the 60% composite

Set	σ_0^W **(MPa)**	σ_1 **(MPa)**	θ_0 **(MPa)**	θ_1 **(MPa)**	**T (°C)**	σ_0^{BMG} **(MPa)**	**n**
1	-1181	-722	738,198	1322	366	-2060	12
% AAE	3.04	1.65	10.3	4.65	15.7	1.88	39
2	-1160	-726	767,685	1317	359	-2079	9
% AAE	2.87	1.64	10.7	4.64	15.5	1.86	41
3	-1150	-727	767,685	1319	360	-2060	10
% AAE	2.97	2.07	10.7	4.64	15.5	1.90	40

Table 6.6 shows the predictions obtained from the 3 sets of inverse neural networks trained for the 60% composite. All predictions are within the FEM ranges. (Table 5.2) Although the %AAE values obtained are relatively high, the neural network predictions are very consistent.

Although presence of additional data points on the 60% experimental stress-strain curve causes a setback for the forward neural network models, it assists the inverse neural network models as the larger dataset provides more information for the prediction of individual parameters. This can be observed in Figures 6.18 through 6.20. Although, the inverse neural network predictions agree well with the experimental data, there is considerable scatter in forward neural network predictions when compared to the other forward models.

Tables 6.7 and 6.8 show the residuals calculated between the experimental and the ANN curves and, the FEM and the ANN curves respectively. Set 2 provides the best match to the experimental data and the residual is very high. Moreover, all parameters are predicted within the FEM ranges. Therefore, further refining of the inverse neural network predictions for this case is not necessary.

Table 6.7 R^2 (residual) values between the experimental and the ANN curves for the 60% composite

Set	**Fiber** R^2	**Composite** R^2	**Average** R^2
1	0.993	0.935	0.9640
2	0.987	0.990	0.9885
3	0.988	0.900	0.9440

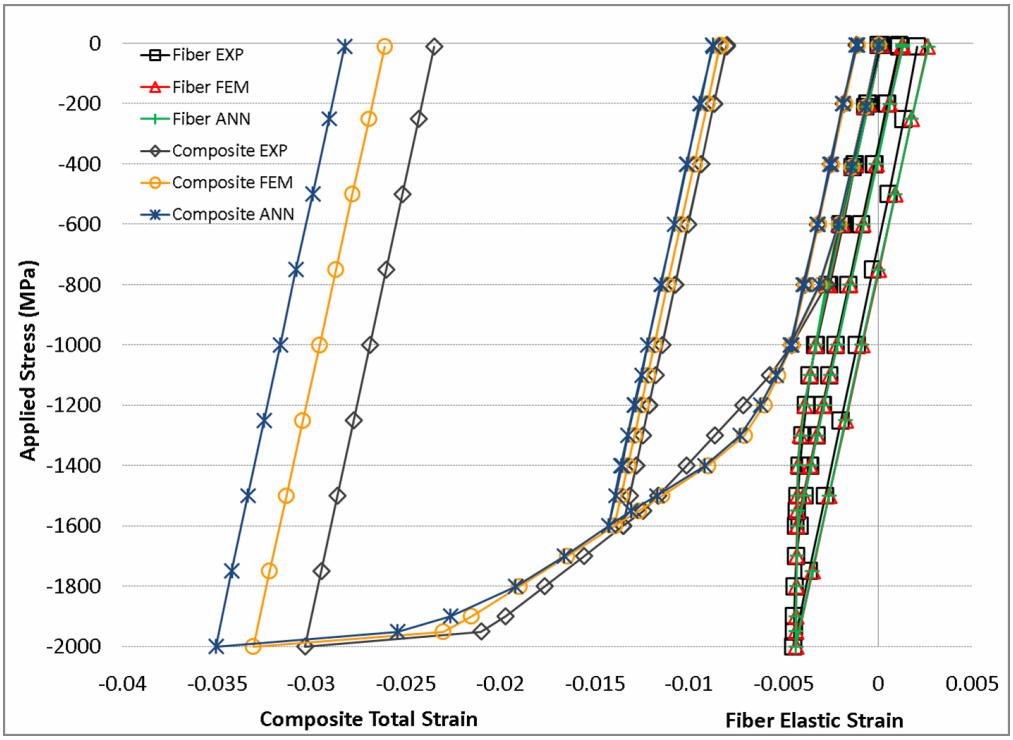


Figure 6.18 Comparison of the experimental, FEM and ANN stress-strain curves for the 60% composite. (Set 1)

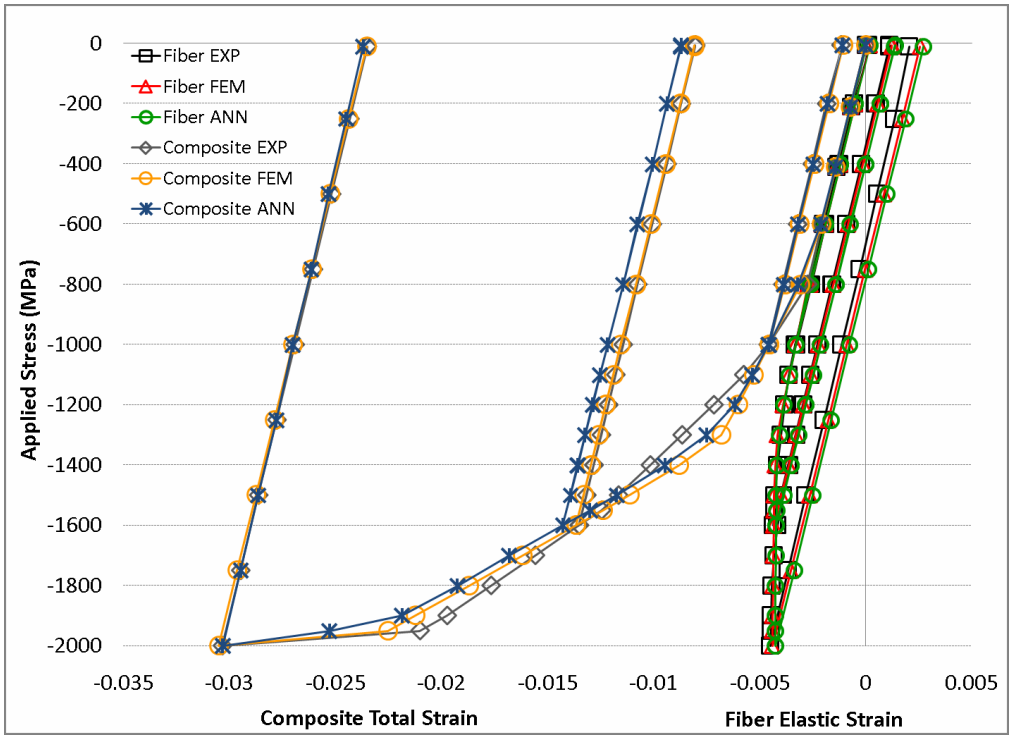


Figure 6.19 Comparison of the experimental, FEM and ANN stress-strain curves for the 60% composite. (Set 2)

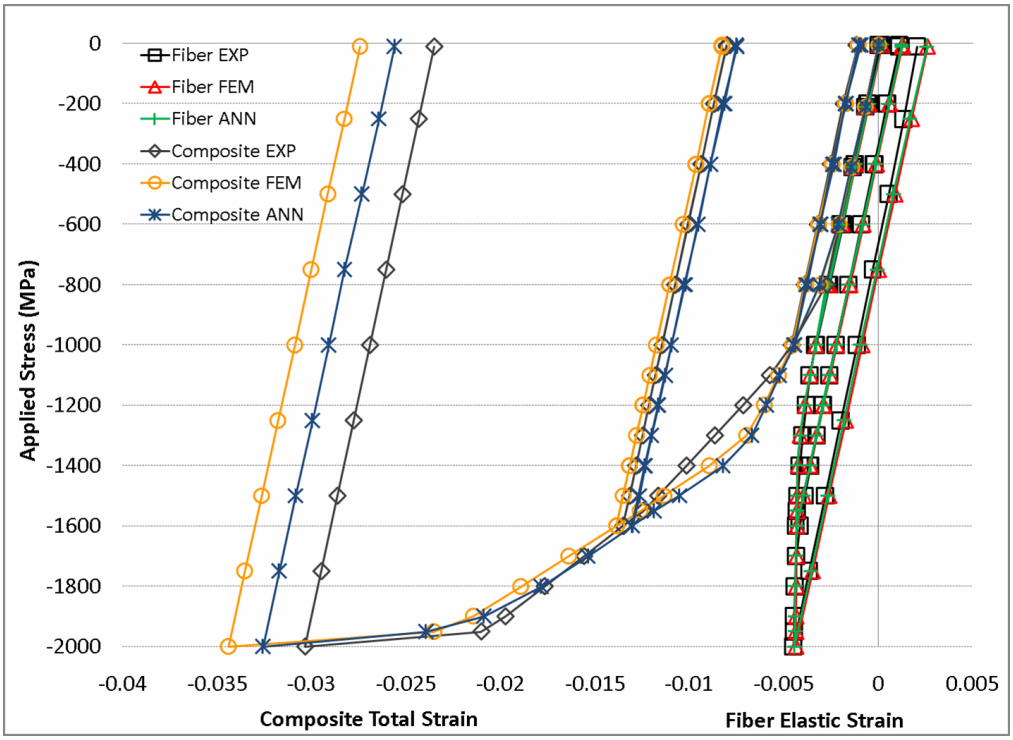


Figure 6.20 Comparison of the experimental, FEM and ANN stress-strain curves for the 60% composite. (Set 3)

Table 6.8 R^2 (residual) values between the FEM and the ANN curves for the 60% composite

Set	**Fiber** R^2	**Composite** R^2	**Average** R^2
1	0.999	0.989	0.9940
2	0.997	0.996	0.9965
3	0.998	0.989	0.9935

6.6.4 80% Composite

Table 6.9 Inverse neural network predictions for the 80% composite

Set	σ_0^W (MPa)	σ_1 (MPa)	θ_0 (MPa)	θ_1 (MPa)	T (°C)	σ_0^{BMG} (MPa)	n
1	-1549	-732	565,698	1331	287	-2007	7
% AAE	1.05	0.40	4.84	4.39	15.6	2.04	39
2	-1542	-722	567,197	1331	279	-2001	8
% AAE	1.04	0.38	4.86	4.43	15.5	2.08	46
3	-1547	-715	564,387	1325	284	-1998	8
% AAE	1.04	0.40	5.29	4.42	15.7	2.20	41

Table 6.9 shows the predictions obtained from the 3 sets of inverse neural networks trained for the 80% composite. Note that, the predictions for σ_0^W and θ_0 are "out-of-range". (Table 5.2) However, the neural network predictions are very consistent and the %AAE values obtained are fairly low for the highly sensitive parameters.

Figures 6.21 through 6.23 show the comparison of the experimental, FEM and ANN stress-strain curves and, Table 6.10 shows the residuals calculated between the experimental and the ANN curves for the 80% composite. Set 2 yielded the highest residual.

There is considerable mismatch between the experimental and the ANN curves. The mismatch mainly results from the "out-of-range" parameters predicted by the inverse neural network models. The inverse neural network predictions for the 80% composite are refined by iterative optimization and the results are presented in the next chapter.

Table 6.11 shows the residuals calculated between the FEM and the ANN curves. Although there is scatter between the forward neural network predictions it is not as significant as the 60% case.

Table 6.10 R^2 (residual) values between the experimental and the ANN curves for the 80% composite

Set	Fiber R^2	Composite R^2	Average R^2
1	0.938	0.936	0.9370
2	0.955	0.934	0.9445
3	0.938	0.936	0.9370

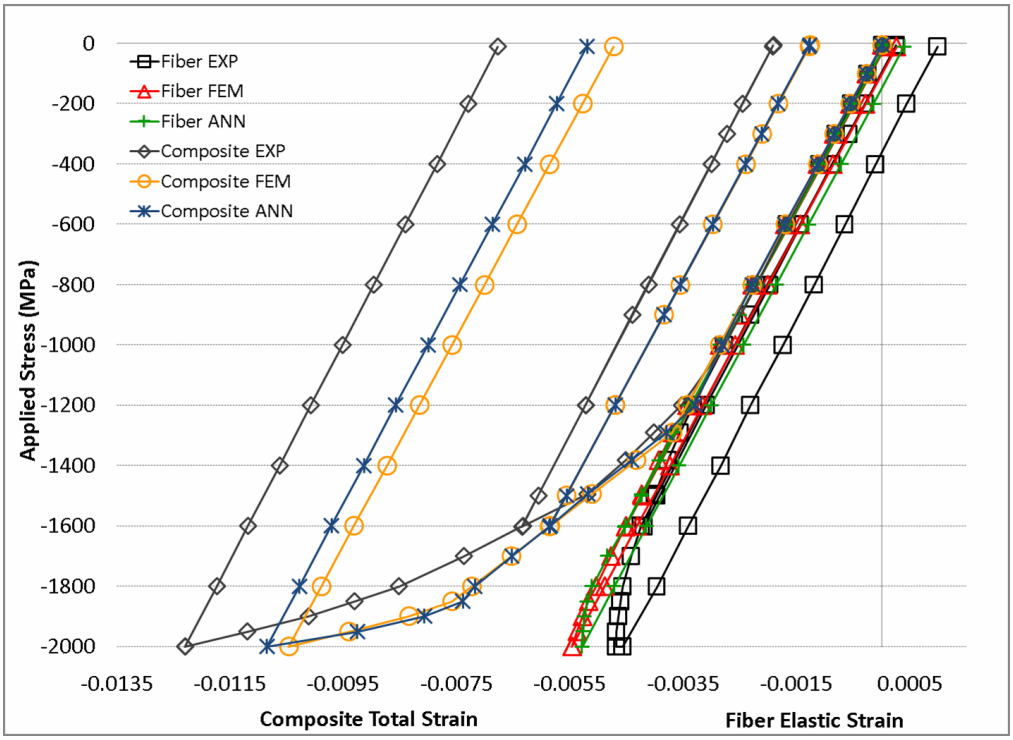


Figure 6.21 Comparison of the experimental, FEM and ANN stress-strain curves for the 80% composite. (Set 1)

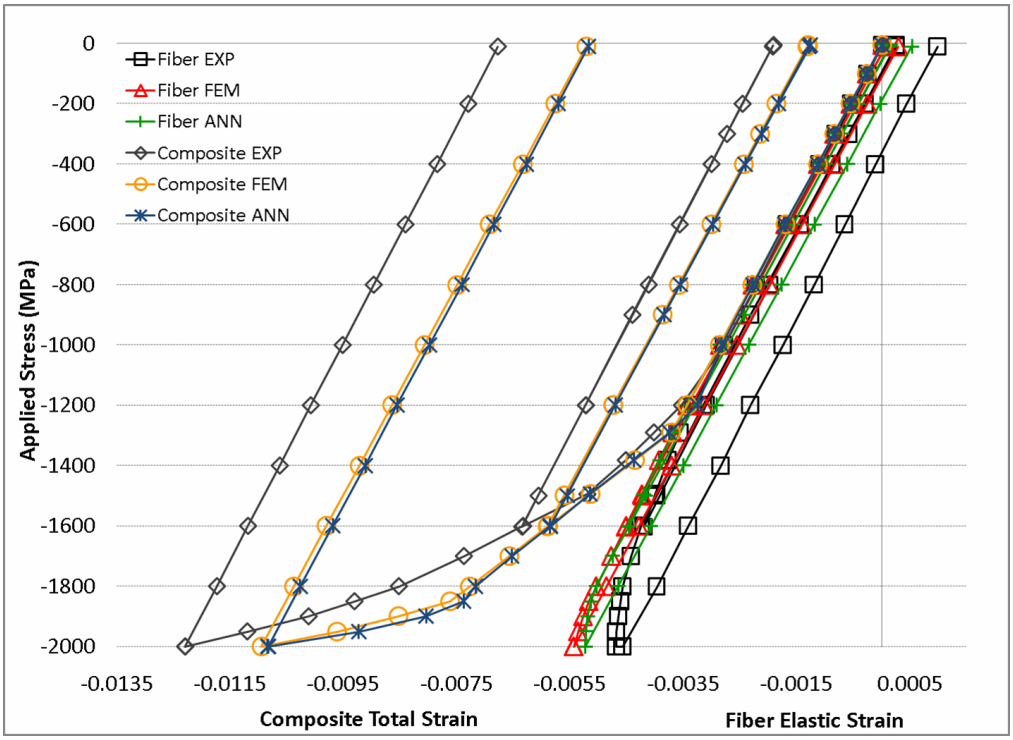


Figure 6.22 Comparison of the experimental, FEM and ANN stress-strain curves for the 80% composite. (Set 2)

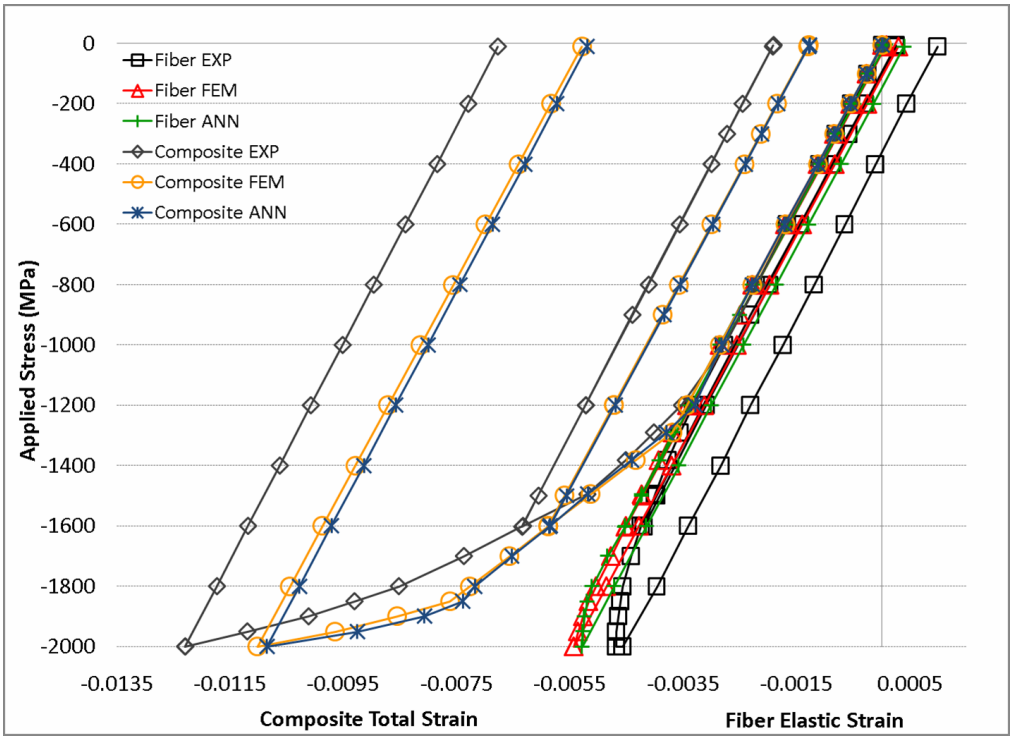


Figure 6.23 Comparison of the experimental, FEM and ANN stress-strain curves for the 80% composite. (Set 3)

Table 6.11 R^2 (residual) values between the FEM and the ANN curves for the 60% composite

Set	**Fiber R^2**	**Composite R^2**	**Average R^2**
1	0.998	0.997	0.9975
2	0.995	0.998	0.9965
3	0.999	0.998	0.9985

6.6.5 Conclusions

The inverse neural network models are used to predict the optimum values of the material parameters that define the constitutive behavior of the fibers and the matrix plus the freezing temperature value with the use of experimental strain data. The predictions compared very well to the experimental data for the 60% composite. Further refinement of the predictions is necessary for the 20%, 40% and 60% composites.

Successful training of inverse neural network models is strongly dependent on the sensitivity of the parameters. The neural networks converged to very low training and testing errors for the highly sensitive parameters. However, limited improvement in neural network predictions is observed for the medium and low sensitivity parameters.

It is shown that the neural network predictions for highly sensitive parameters are significantly affected by the changes in neural network architecture.

The initial array of connection weights has a significant effect on the neural network outcomes. For instance the highest residuals between the experimental and the ANN-predicted curves are obtained for the set 2 for all composites. This result suggests that the initial connection weights employed for this set provided a steeper decent in the prediction errors.

The mismatch between the experimental and the ANN-predicted curves mainly caused by prediction of "out-of-range" parameters by the inverse neural network models. The main reasons leading to this result are limitations of the inverse neural network models in predicting medium and low sensitivity parameters and the simplifying assumptions employed during development of the finite element models which, are used to generate the training and testing datasets for the neural network analysis.

CHAPTER 7. ITERATIVE OPTIMIZATION

In this study, the aim of iterative optimization is to refine the inverse neural network predictions for the "out-of-range" parameters. This way the optimum values of "out-of-range" parameters are found and the match between the experimental and the ANN-predicted stress-strain curves is improved. In optimizing the parameters the results of the sensitivity study (Chapter 4) are consulted to make educated decisions on the iteration intervals and sometimes on the additional parameters that are required to be optimized.

The value of the optimized parameter is varied within an interval at discrete points while the other sensitive parameters governing the constitutive behavior of the composites are kept constant. Here, it should be noted that, there are multiple solutions i.e., different set of numbers, yielding very similar curves for each composite. Therefore, special attention is given to preservation of the inverse neural network predictions and the optimum value of the parameter is obtained with the lowest possible number of iterations.

The iterations are started with the predictions of set 2 for each composite which, consistently yielded the highest residuals between the experimental and the ANN-predicted curves for each composite. (Chapter 6) The forward neural network models developed (Chapter 5) are used to obtain the predictions and speed-up the optimized iteration process. This is facilitated by presenting the discrete points tested in each interval along with the constant values of other parameters as inputs to the forward neural network models in a testing set and the corresponding stress-strain curves are obtained for each iteration.

7.1 20% Composite

The mismatch between the experimental and the ANN-predicted stress-strain curves for the 20% composite is concentrated at the final unloading portion of the curve. (Figure 6.13) To be able to improve the fit between the two curves a parameter that is only effective during that portion is needed. The inverse neural network models for the 20% composite predicted σ_0^W, σ_1 and σ_0^{BMG} outside the ranges of the FEM training datasets. (Table 6.2) According to the sensitivity study results only σ_0^{BMG} is effective at that specific portion of the curve among the three "out-of-range" parameters. Consequently, the value of σ_0^{BMG} should be optimized for refining the inverse neural network predictions.

Figure 7.1 shows the first pass of the iterative optimization where, σ_0^{BMG} is varied by 50 MPa intervals between -2050 and -1900 MPa . ANN ORG refers to the original inverse neural network prediction obtained from the second set of inverse neural networks before iterative optimization. For the second pass of optimization the interval is narrowed down to $[-2000, -1960]$ and σ_0^{BMG} is varied by 10 MPa intervals. The stress-strain curves are presented on Figure 7.2. The best match is obtained at $\sigma_o^{BMG} = -1980$ MPa. (Figure 7.3) The residuals between the experimental and the ANN-predicted curves before and after the iterative optimization is presented on Table 7.1.

Table 7.1 R^2 (residual) values between the experimental and the ANN-predicted curves before and after iterative optimization for the 20% composite

Curve	**Fiber** R^2	**Composite** R^2	**Average** R^2
Original	0.971	0.979	0.9750
Optimized	0.970	0.996	0.9830

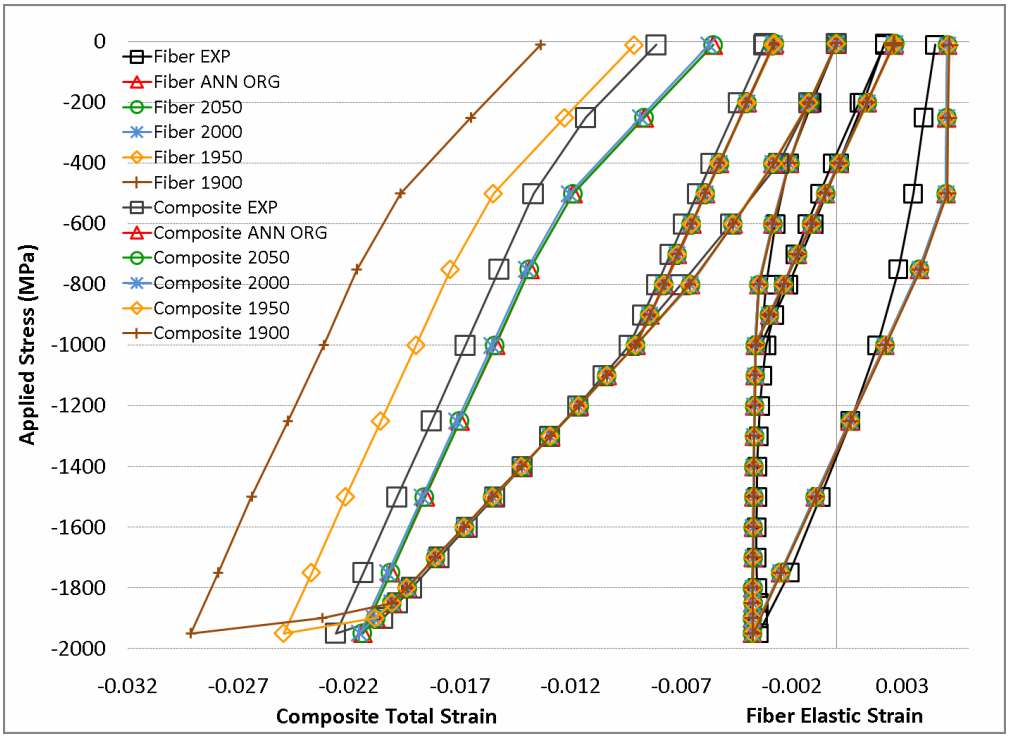


Figure 7.1 Comparison of stress-strain curves for the first pass of iterative optimization for the 20% composite

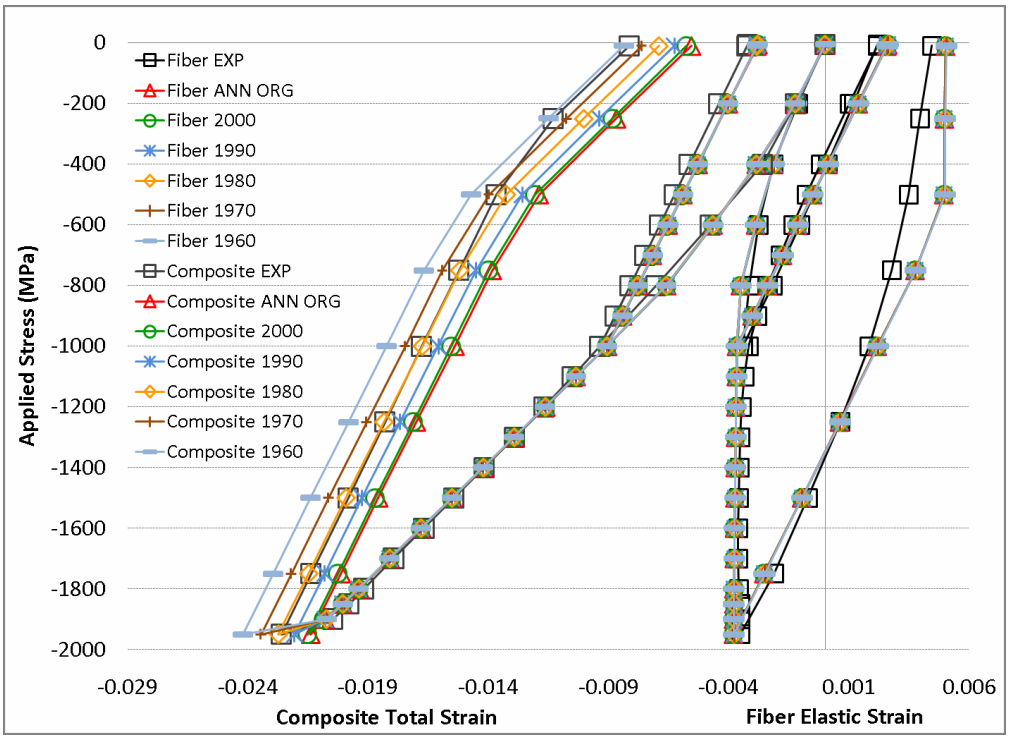


Figure 7.2 Comparison of stress-strain curves for the second pass of iterative optimization for the 20% composite

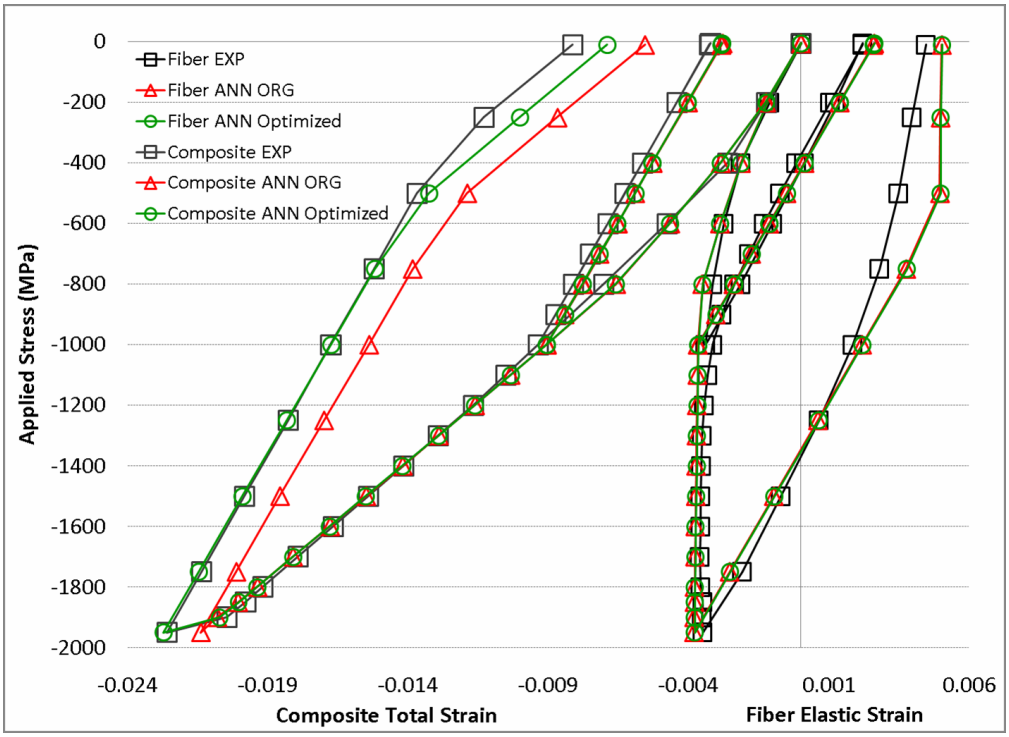


Figure 7.3 Comparison of stress-strain curves after iterative optimization for the 20% composite

7.2 40% Composite

The mismatch between the experimental and ANN-predicted stress-strain curves for the 40% composite is more pronounced than that of the 20% case. The mismatch is caused by the "out-of-range" prediction of σ_0^W. A three stage iterative optimization is employed to improve the fit between the experimental and the ANN curves. For the first pass of optimization σ_0^W is varied between -1450 and -1150 MPa at 100 MPa intervals. (Figure 7.4). Then the interval is narrowed down to $[-1350, -1250]$ and σ_0^W is varied with 25 MPa intervals. (Figure 7.5) Finally, the interval is narrowed down to $[-1290, -1280]$ and the best match is obtained at $\sigma_0^W = -1285$ MPa. The residuals between the experimental and the ANN-predicted curves before and after the iterative optimization is presented on Table 7.2.

Table 7.2 R^2 (residual) values between the experimental and the ANN-predicted curves before and after iterative optimization for the 40% composite

Curve	**Fiber R^2**	**Composite R^2**	**Average R^2**
Original	0.971	0.874	0.9225
Optimized	0.974	0.998	0.9860

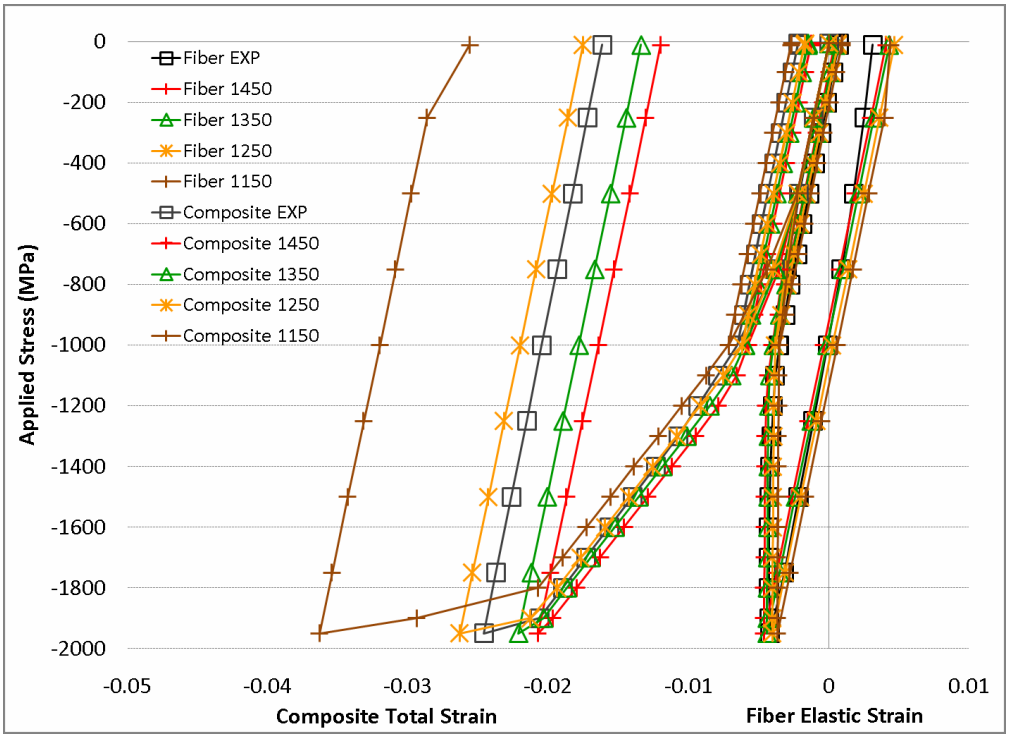


Figure 7.4 Comparison of stress-strain curves for the first pass of iterative optimization for the 40% composite

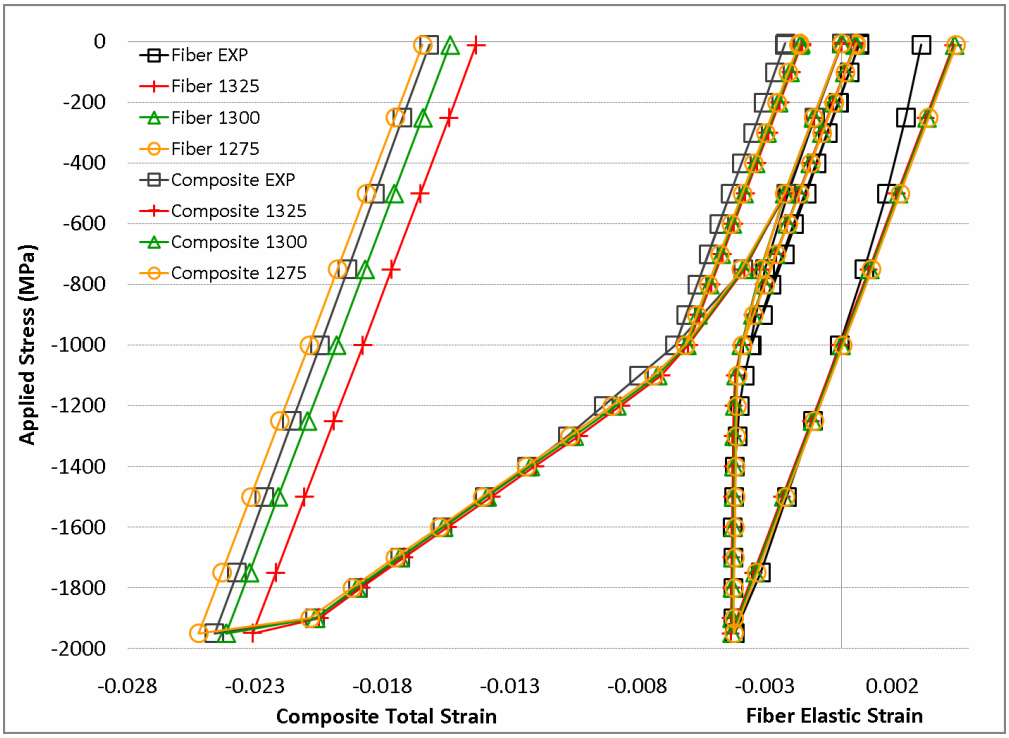


Figure 7.5 Comparison of stress-strain curves for the second pass of iterative optimization for the 40% composite

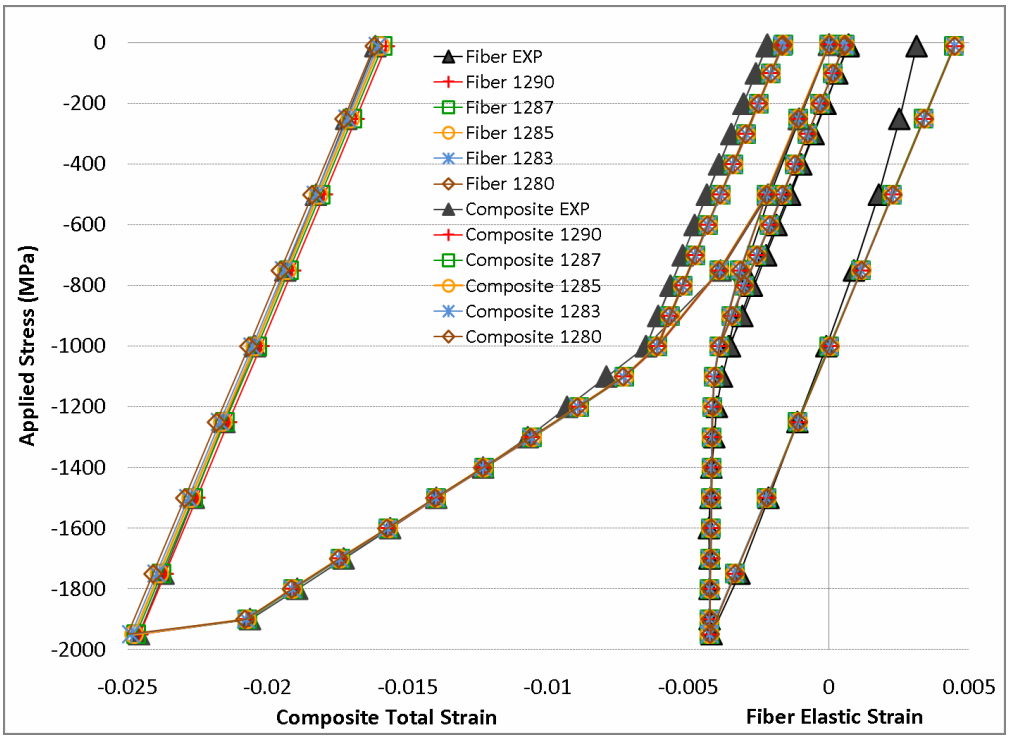


Figure 7.6 Comparison of stress-strain curves for the third pass of iterative optimization for the 40% composite

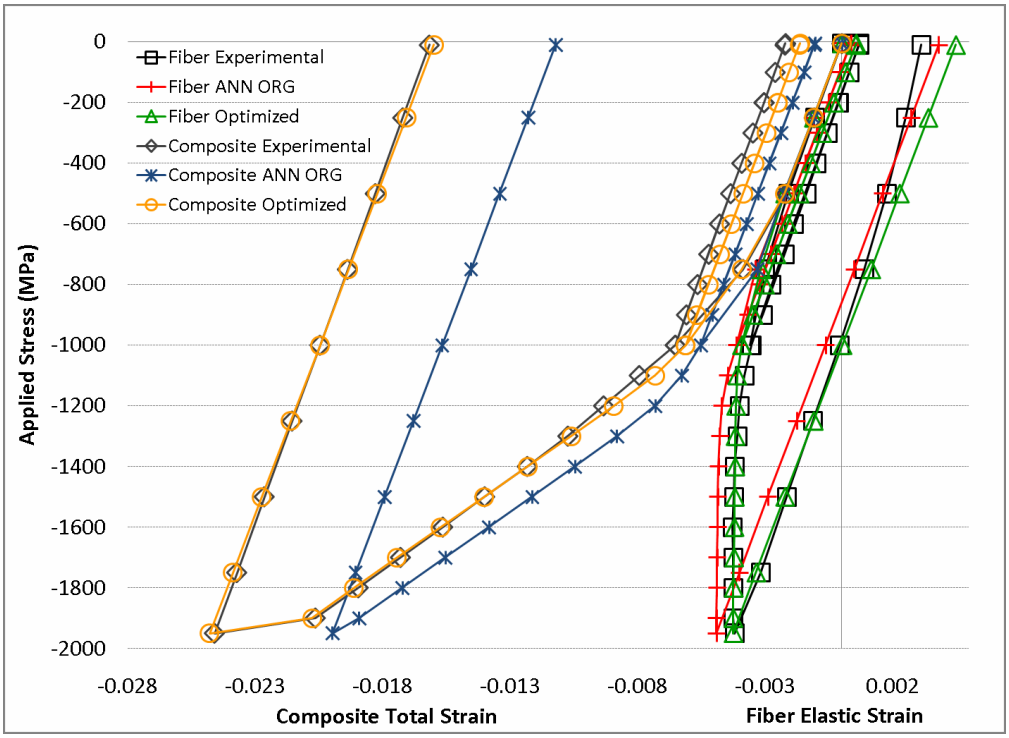


Figure 7.7 Comparison of stress-strain curves after iterative optimization for the 40% composite

7.3 80% Composite

The mismatch between the experimental and the ANN-predicted stress-strain curves for the 80% composite is the most complex of the cases studied due to the considerable misfit during both the first and the second unloading portions of the curve. Inverse neural network models yielded "out-of-range" values for σ_0^W and θ_0. However, σ_0^W is the only parameter with considerable sensitivity during the first unloading portion of the curve. When the mismatch at the first unloading portion is fixed by varying σ_0^W, it increases the mismatch during the second unloading part. Moreover, θ_0 is not sensitive enough to account for the induced mismatch at the second unloading portion. The results of the sensitivity study suggest optimizing σ_1 - the second most sensitive parameter. However, σ_1 has a minor sensitivity at the first unloading portion of the curve. This makes the problem even more complicated. Therefore, refinement of inverse neural network models for the 80% composite requires a more comprehensive

optimization than the 20% and 40% composites.

Figure 7.8 summarizes the approach undertaken. First σ_0^W is varied between -1450 and -1250 MPa with 50 MPa intervals and the best fit is obtained at $\sigma_0^W = -1450$ MPa. Then, σ_1 is varied between -750 and -800 MPa with 25 MPa intervals and the best match is obtained at $\sigma_1 = -775$ MPa. Afterwards, θ_0 is varied between 600,000 and 670,000 MPa with 35,000 MPa intervals. However, the variation in θ_0 did not yield to a noticeable difference in the stress-strain curve. The iterative optimization is continued with $\theta_0 = 635,000$. (mean value of the employed interval) Then, σ_0^W is further optimized in two more passes. At the first pass it is varied between -1450 and -1400 MPa with 10 MPa intervals and the best result is obtained at $\sigma_0^W = -1430$ MPa. Finally, it is varied between -1438 and -1434 MPa with 1 MPa intervals and the best match is obtained at $\sigma_0^W = -1435$ MPa. (Figure 7.9) The residuals between the experimental and the ANN-predicted curves before and after the iterative optimization is presented on Table 7.3.

Table 7.3 R^2 (residual) values between the experimental and the ANN-predicted curves before and after iterative optimization for the 80% composite

Curve	**Fiber** R^2	**Composite** R^2	**Average** R^2
Original	0.955	0.934	0.9445
Optimized	0.996	0.986	0.9910

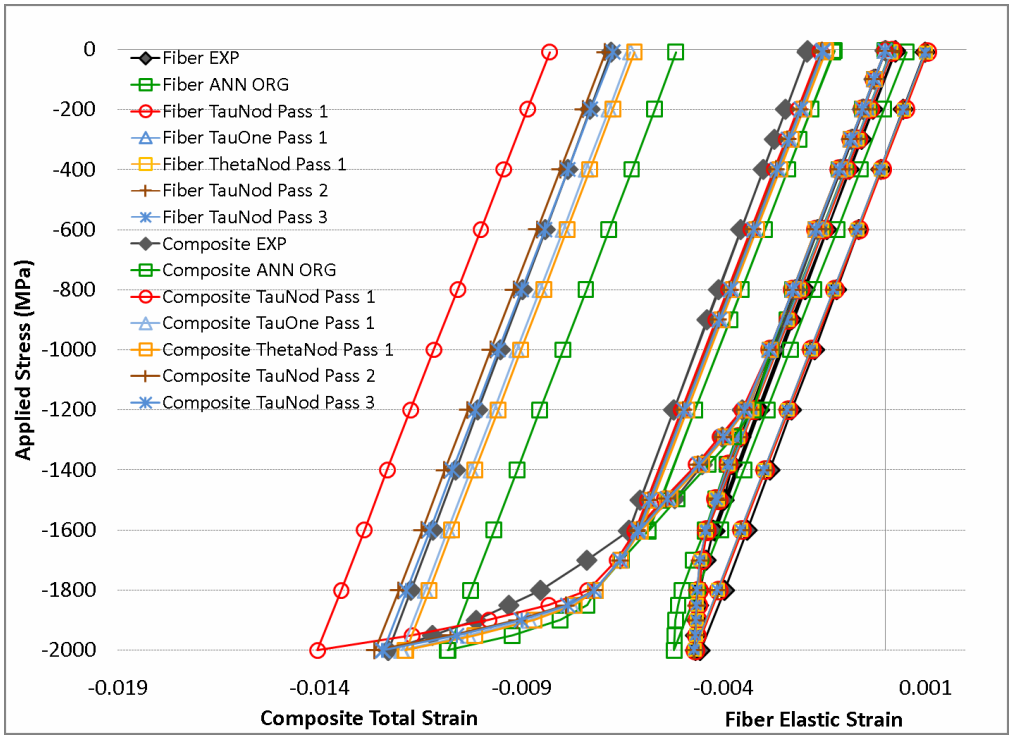


Figure 7.8 Comparison of stress-strain curves for the passes of iterative optimization for the 80% composite

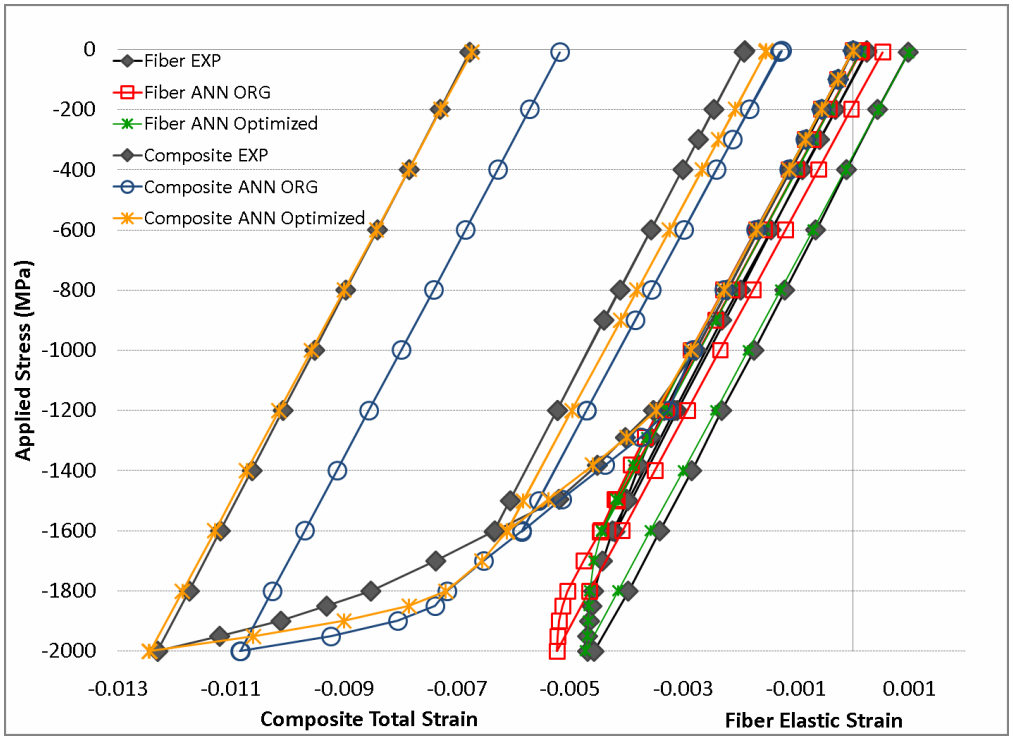


Figure 7.9 Comparison of stress-strain curves after iterative optimization for the 80% composite

7.4 Conclusions

Iterative optimization yielded significant improvements in residuals between the experimental and ANN-predicted stress-strain curves for the 20%, 40% and 80% composites. (Tables 7.1 - 7.3) Iterative optimization is not applied to the 60% composite because the inverse neural network models consistenly predicted the parameters within the FEM ranges (Table 6.6) and the fit residuals obtained are very good. (Table 6.7)

Tables 7.4 through 7.6 show the inverse neural network model predictions and the values of the parameters after iterative optimization.

Table 7.4 Inverse neural network predictions before and after iterative optimization for the 20% composite

Prediction	σ_0^W **(MPa)**	σ_1 **(MPa)**	**T (°C)**	σ_0^{BMG} **(MPa)**	**n**
Original	-938	-825	318	-2105	8
Optimized	-938	-825	318	-1980	8

Table 7.5 Inverse neural network predictions before and after iterative optimization for the 40% composite

Prediction	σ_0^W **(MPa)**	σ_1 **(MPa)**	**T (°C)**	σ_0^{BMG} **(MPa)**	**n**
Original	-1561	-560	316	-1965	6
Optimized	-1285	-560	316	-1965	6

Table 7.6 Inverse neural network predictions before and after iterative optimization for the 80% composite

Prediction	σ_0^W **(MPa)**	σ_1 **(MPa)**	θ_0 **(MPa)**	θ_1 **(MPa)**	**T (°C)**	σ_0^{BMG} **(MPa)**	**n**
Original	-1542	-722	567,197	1331	279	-2001	8
Optimized	-1435	-775	635,000	1331	279	-2001	8

CHAPTER 8. CONCLUSIONS

The results of the sensitivity study proved to be instrumental in developing forward and inverse neural network models, and building the iterative optimization approach for the refinement of inverse neural network predictions. The sensitivity of the parameters are found to depend on fiber volume fraction. Voce law parameters θ_0 and θ_1 are found to be insensitive for the 20% and 40% composites. This shows that Voce law has limitations in modeling the in-situ constitutive behavior of tungsten.

All forward neural network models successfully predicted the constitutive behavior of the composites accurately in shorter times than estimated by the FEM. The forward neural network models are also integrated in a framework to validate the inverse neural network predictions and to perform faster iterative optimization.

The inverse neural network predictions compared very well to the experimental data for the 60% composite, however, refinement of the predictions are required for the 20%, 40% and 80% composites. For these composites mentioned the inverse neural network models predicted some of the parameters outside the ranges employed by the FEM. This counter-intuitive outcome is attributed to the limitations of the neural network models in predicting low and medium sensitivity parameters and the simplifying assumtions made during the development of the FEM, which is used to generate the database for training the neural network models.

The iterative optimization approach adopted improved the match between the inverse neural network predictions and the experimental data significantly for the 20%, 40% and 80% composites. The forward neural network models played a crucial role in implementing the iterative optimization process.

BIBLIOGRAPHY

[1] S. Haykin. *Neural networks; a comprehensive foundation, 2nd Edition.* Prentice-Hall, Upper Saddle River, NJ, 1999.

[2] S.-Y. Lee. *Deformation mechanisms of bulk metallic glass matrix composites.* PhD thesis, California Institute of Technology, Pasadena, CA, 2005.

[3] E. Voce. The relationship between stress and strain for homogeneous deformation. *Journal of the institute of metals*, 74:534–562, 1948.

[4] E. Ustundag. Unpublished research, 2005.

[5] D. Dragoi, E. Ustundag, B. Clausen, and M.A.M Bourke. Investigation of thermal residual stresses in tungsten-fiber/bulk metallic glass composites. *Scripta Materialia*, 45:245–252, 2001.

[6] B. Clausen, S.-Y. Lee, E. Ustundag, C.C. Aydiner, R.D. Conner, and M.A.M Bourke. Compressive yielding of tungsten fiber reinforced bulk metallic glass composites. *Scripta Materialia*, 49:123–128, 2003.

[7] Krogh A. Palmer R. G. Hertz, J. *Introduction to the theory of neural computation.* Addison-Wesley, Redwood City, CA, 1991.

[8] C. Lau. *Neural networks, theoretical foundations and analysis.* IEEE Press, New York, NY, 1992.

[9] D. Graupe. *Principles of Artificial neural networks, 2nd Edition.* World Scientific, Singapore, 2007.

[10] L. Fausett. *Fundamentals of neural networks; architectures, algorithm, and applications*. Prentice-Hall, Englewood Cliffs, NJ, 1994.

[11] Taylor J.G. *Neural networks*. Unicom Ltd., Oxon, UK, 1995.

[12] Obradovic D. Deco, G. *An information-theoretic approach to neural computing*. Springer-Verlag, New York, NY, 1996.

[13] McClelland J. L. Rumelhart, D. E. and the PDP Research Group. *Parallel distributed processing*. MIT Press, Cambridge, MA, 1986.

[14] T. L. Fine. *Feedforward neural network methodology*. Springer-Verlag, New York, NY, 1999.

[15] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986.

[16] S. J. Joyce, D. J. Osguthorpe, J. A. Padgett, and G. J. Price. Neural-network prediction of glass-transition temperatures from monomer structure. *Journal of the Chemical Society-Faraday Transactions*, 91(16):2491–2496, 1995.

[17] T. R. Cundari and E. W. Moody. A comparison of neural networks versus quantum mechanics for inorganic systems. *Journal of Chemical Information and Computer Sciences*, 37(5):871–875, 1997.

[18] Y. Asada, E. Nakada, S. Matsumoto, and H. Uesaka. Prediction of t-c for yba2cu3oz doped with ca using neural network. *Journal of Superconductivity*, 10(1):23–26, 1997.

[19] S. Malinov, W. Sha, and Z. Guo. Application of artificial neural network for prediction of time-temperature-transformation diagrams in titanium alloys. *Materials Science and Engineering a-Structural Materials Properties Microstructure and Processing*, 283(1-2):1–10, 2000.

[20] S. Malinov, W. Sha, and J. J. McKeown. Modelling the correlation between processing parameters and properties in titanium alloys using artificial neural network. *Computational Materials Science*, 21(3):375–394, 2001.

[21] J. McBride, S. Malinov, and W. Sha. Modelling tensile properties of gamma-based titanium aluminides using artificial neural network. *Materials Science and Engineering a-Structural Materials Properties Microstructure and Processing*, 384(1-2):129–137, 2004.

[22] S. McShane, S. Malinov, J.J. McKeown, and W. Sha. Simulation of fatigue stress life (s-n) diagrams for ti-6al-4v alloy by application of artificial neural network. pages 653 – 662, San Diego, CA, United States, 2001.

[23] S. Malinov and W. Sha. Application of artificial neural networks for modelling correlations in titanium alloys. *Materials Science and Engineering a-Structural Materials Properties Microstructure and Processing*, 365(1-2):202–211, 2004.

[24] D. J. C. Mackay. A practical bayesian framework for backpropagation networks. *Neural Computation*, 4(3):448–472, 1992.

[25] L. Gavard, Hkdh Bhadeshia, D. J. C. MacKay, and S. Suzuki. Bayesian neural network model for austenite formation in steels. *Materials Science and Technology*, 12(6):453–463, 1996.

[26] W. G. Vermeulen, P. F. Morris, A. P. deWeijer, and S. vanderZwaag. Prediction of martensite start temperature using artificial neural networks. *Ironmaking Steelmaking*, 23(5):433–437, 1996.

[27] W. G. Vermeulen, P. J. vanderWolk, A. P. deWeijer, and S. vanderZwaag. Prediction of jominy hardness profiles of steels using artificial neural networks. *Journal of Materials Engineering and Performance*, 5(1):57–63, 1996.

[28] W. Vermeulen, S. vanderZwaag, P. Morris, and T. deWeijer. Prediction of the continuous cooling transformation diagram of some selected steels using artificial neural networks. *Steel Research*, 68(2):72–79, 1997.

[29] Hkdh Bhadeshia, D. J. C. MacKay, and L. E. Svensson. Impact toughness of c-mn steel arc welds - bayesian neural network analysis. *Materials Science and Technology*, 11(10):1046–1051, 1995.

[30] Kazutoshi Ichikawa, H. K. D. H. Bhadeshia, and D. J. C. MacKay. Model for solidification cracking in low alloy steel weld metals. *Science and Technology of Welding and Joining*, 1(1), 1996.

[31] Tracey Cool, H. K. D. H. Bhadeshia, and D. J. C. MacKay. The yield and ultimate tensile strength of steel welds. *Materials Science and Engineering: A*, 223(1-2):186–200, 1997.

[32] S. Yoshitake, V. Narayan, H. Harada, Hkdh Bhadeshia, and D. J. C. Mackay. Estimation of the gamma and gamma ' lattice parameters in nickel-base superalloys using neural network analysis. *Isij International*, 38(5):495–502, 1998.

[33] H. Fujii, D. J. C. Mackay, and Hkdh Bhadeshia. Bayesian neural network analysis of fatigue crack growth rate in nickel base superalloys. *Isij International*, 36(11):1373–1382, 1996.

[34] J. M. Schooling, M. Brown, and P. A. S. Reed. An example of the use of neural computing techniques in materials science - the modelling of fatigue thresholds in ni-base superalloys. *Materials Science and Engineering a-Structural Materials Properties Microstructure and Processing*, 260(1-2):222–239, 1999.

[35] S. B. Singh, Hkdh Bhadeshia, D. J. C. MacKay, H. Carey, and I. Martin. Neural network analysis of steel plate processing. *Ironmaking Steelmaking*, 25(5):355–365, 1998.

[36] P. Korczak, H. Dyja, and E. Labuda. Using neural network models for predicting mechanical properties after hot plate rolling processes. *Journal of Materials Processing Technology*, 80-1:481–486, 1998.

[37] W. Vermeulen, A. Bodin, and S. vanderZwaag. Prediction of the measured temperature after the last finishing stand using artificial neural networks. *Steel Research*, 68(1):20–26, 1997.

[38] J. Larkiola, P. Myllykoski, A. S. Korhonen, and L. Cser. The role of neural networks in the optimisation of rolling processes. *Journal of Materials Processing Technology*, 80-1:16–23, 1998.

[39] Z. Y. Liu, W. D. Wang, and W. Gao. Prediction of the mechanical properties of hot-rolled c-mn steels using artificial neural networks. *Journal of Materials Processing Technology*, 57(3-4):332–336, 1996.

[40] Hkdh Bhadeshia. Neural networks in materials science. *Isij International*, 39(10):966–979, 1999.

[41] S. M. Roberts, J. Kusiak, Y. L. Liu, A. Forcellese, and P. J. Withers. Prediction of damage evolution in forged aluminium metal matrix composites using a neural network approach. *Journal of Materials Processing Technology*, 80-1:507–512, 1998.

[42] N. Rai and R. Pitchumani. Rapid cure simulation using artificial neural networks. *Composites Part a-Applied Science and Manufacturing*, 28(9-10):847–859, 1997.

[43] H. B. Su, L. T. Fan, and J. R. Schlup. Monitoring the process of curing of epoxy/graphite fiber composites with a recurrent neural network as a soft sensor. *Engineering Applications of Artificial Intelligence*, 11(2):293–306, 1998.

[44] Y. Al-Assaf and H. El Kadi. Fatigue life prediction of unidirectional glass fiber/epoxy composite laminae using neural networks. *Composite Structures*, 53(1):65–71, 2001.

[45] Hany El Kadi and Yousef Al-Assaf. Prediction of the fatigue life of unidirectional glass fiber/epoxy composite laminae using different neural network paradigms. *Composite Structures*, 55(2):239–246, 2002.

[46] Z. Zhang, N. M. Barkoula, J. Karger-Kocsis, and K. Friedrich. Artificial neural network predictions on erosive wear of polymers. *Wear*, 255:708–713, 2003.

[47] Z. Zhang, K. Friedrich, and K. Velten. Prediction on tribological properties of short fibre composites using artificial neural networks. *Wear*, 252(7-8):668–675, 2002.

[48] Z. Zhang, P. Klein, and K. Friedrich. Dynamic mechanical properties of ptfe based short carbon fibre reinforced composites: experiment and artificial neural network prediction. *Composites Science and Technology*, 62(7-8):1001–1009, 2002.

[49] Z. Zhang and K. Friedrich. Artificial neural networks applied to polymer composites: a review. *Composites Science and Technology*, 63(14):2029–2044, 2003.

[50] H. S. Rao and A. Mukherjee. Artificial neural networks for predicting the macromechanical behaviour of ceramic-matrix composites. *Computational Materials Science*, 5(4):307–322, 1996.

[51] R. Haj-Ali, D. A. Pecknold, J. Ghaboussi, and G. Z. Voyiadjis. Simulated micromechanical models usings artificials neural networks. *Journal of Engineering Mechanics*, 127(7):730–738, 2001.

[52] A. Mukherjee, S. Schmauder, and M. Ruhle. Artificial neural networks for the prediction of mechanical behavior of metal matrix composites. *Acta Metallurgica et Materialia*, 43(11):4083–4091, 1995.

[53] H. Rao, Vaishali G. Ghorpade, and A. Mukherjee. A genetic algorithm based back propagation network for simulation of stress-strain response of ceramic-matrix-composites. *Computers and Structures*, 84(5-6):330–339, 2006.

[54] Rami Haj-Ali and Hoan-Kee Kim. Nonlinear constitutive models for frp composites using artificial neural networks. *Mechanics of Materials*, 39(12):1035–1042, 2007.

[55] N. Huber and C. Tsakmakis. Determination of constitutive properties from spherical indentation data using neural networks. part i: the case of pure kinematic hardening in plasticity laws. *Journal of the Mechanics and Physics of Solids*, 47(7):1569–1588, 1999.

[56] N. Huber and C. Tsakmakis. Determination of constitutive properties from spherical indentation data using neural networks. part ii: plasticity with nonlinear isotropic and kinematic hardening. *Journal of the Mechanics and Physics of Solids*, 47(7):1589–1607, 1999.

[57] N. Huber, A. Konstantinidis, and C. Tsakmakis. Determination of poisson's ratio by spherical indentation using neural networks - part i: Theory. *Journal of Applied Mechanics-Transactions of the Asme*, 68(2):218–223, 2001.

[58] N. Huber and C. Tsakmakis. Determination of poisson's ratio by spherical indentation using neural networks - part ii: Identification method. *Journal of Applied Mechanics-Transactions of the Asme*, 68(2):224–229, 2001.

[59] Anastasia Muliana, Rejanah Steward, Rami M. Haj-Ali, and Ashok Saxena. Artificial neural network and finite element modeling of nanoindentation tests. *Metallurgical and Materials Transactions A: Physical Metallurgy and Materials Science*, 33(7):1939–1947, 2002.

[60] N. Huber, W. D. Nix, and H. Gao. Identification of elastic-plastic material parameters from pyramidal indentation of thin films. *Proceedings of the Royal Society of London Series a-Mathematical Physical and Engineering Sciences*, 458(2023):1593–1620, 2002.

[61] N. Huber, I. Tsagrakis, and C. Tsakmakis. Determination of constitutive properties of thin metallic films on substrates by spherical indentation using neural networks. *International Journal of Solids and Structures*, 37(44):6499–6516, 2000.

[62] H. Ceylan. *Analysis and design of concrete pavement systems using artificial neural networks*. PhD thesis, University of Illinois at Urbana-Champaign, Urbana, IL, 2002.